



UNIVERSIDAD CARLOS III DE MADRID

Ingeniería Informática

**DESARROLLO DE UNA APLICACIÓN SOFTWARE
PARA LABORATORIOS REMOTOS: CONTROL REMOTO DE
UN MOTOR**

TRABAJO DE FIN DE GRADO

Madrid, Febrero 2014

Autora: Geisel Zayas Sierra

Tutor: Javier Fernández Muñoz

Resumen

El presente documento describe el proceso de desarrollo de una aplicación web para la realización de prácticas. La práctica concreta que aborda este proyecto consiste en la visualización de un motor localizado en una ubicación remota, y el envío y recepción de datos a una herramienta que controla dicho motor. La aplicación permite, a los estudiantes registrados en su base de datos, la reserva, acceso e interacción con las herramientas que permiten visualizar e interactuar con el motor, ubicadas en un laboratorio remoto.

Abstract

The present document describes the process of developing a web application for student practices. The practice that this project addresses consists of visualization of an engine in a remote location, and sending and receiving data to/from the tool controlling the engine. The application allows the students registered in its data base to book, access and interact with the tools (located in a remote laboratory) controlling the visualization and interaction with the engine.

ÍNDICE DE CONTENIDO

1. Introducción.....	21
1.1. Visión general	22
1.2. Motivación	23
1.3. Objetivos.....	24
1.4. Fases de desarrollo	25
1.5. Contenido del documento.....	25
2. Estado del arte.....	27
2.1. Evolución de la tecnología	28
2.1.1. Comunicaciones.....	28
2.1.2. Almacenamiento	31
2.2. Tecnología utilizada	32
2.2.1. Entorno de desarrollo.....	32
2.2.2. Lenguajes de desarrollo.....	33
2.2.2.1. <i>Java</i>	33
2.2.2.2. <i>JavaScript</i>	34
2.2.2.3. <i>HTML</i>	37
2.2.2.4. <i>CSS</i>	37
2.2.2.5. <i>Shell script</i>	37
2.2.3. Servidor de aplicaciones.....	38
2.2.4.2. <i>SQLite</i>	39
2.2.5. Modelo de trabajo.....	40
2.2.6. Comunicaciones.....	42
2.2.6.1. <i>Sockets</i>	42

3. Análisis de la solución	47
3.1. Descripción de la solución	48
3.1.1. Capacidades generales	48
3.1.2. Restricciones generales	48
3.1.3. Características de los usuarios	49
3.1.4. Entorno operacional	49
3.2. Análisis de requisitos	49
3.2.1. Requisitos de usuario	49
3.2.1.1. <i>Requisitos de capacidad</i>	51
3.2.1.2. <i>Requisitos de restricción</i>	68
3.2.2. Casos de uso	75
3.2.3. Requisitos de software	91
3.2.3.1 <i>Requisitos funcionales</i>	92
3.2.3.2 <i>Requisitos no funcionales</i>	119
3.2.4. Matriz de trazabilidad.....	123
4. Diseño	127
4.1. Arquitectura del sistema	128
4.2. Base de datos.....	130
4.3. Interfaces de usuario	133
4.3.1. Inicio de sesión	134
4.3.2. Listado de reservas	135
4.3.3. Calendario de reservas	139
4.3.4. Ayuda	142
4.3.5. Cuestionario.....	143

4.3.6. Configuración del motor.....	145
4.3.7. Carga de la reserva	147
4.3.8. Ejecución de la reserva	148
4.4. Diagrama de navegación	153
5. Implementación.....	157
5.1. Servidor.....	158
5.2. Herramientas	169
5.3. Diagrama de despliegue	170
6. Pruebas	173
6.1. Pruebas de funcionalidad	174
6.2. Matriz de trazabilidad.....	184
7. Planificación.....	187
7.1. Planificación de actividades.....	188
7.2. Diagrama de Gantt	188
8. Presupuesto	191
8.1. Costes de desarrollo	192
8.1.1. Costes de personal	192
8.1.2. Costes materiales	192
8.1.3. Coste total	193
8.2. Costes de despliegue	193
8.2.1. Costes del servidor	194
8.2.2. Costes del laboratorio	194
8.2.3. Costes de las herramientas	194
8.2.4. Coste total	195

9. Conclusiones	197
9.1. Conclusiones generales	198
9.2. Conclusiones personales	199
9.3. Trabajos futuros.....	200
Anexo I: Guía de usuario.....	203
Anexo II: Guía de despliegue	215
Glosario de términos y siglas	221
Bibliografía.....	225

ÍNDICE DE TABLAS

Tabla 1: Requisito de usuario 50

Tabla 2: Requisito RU-C-01 51

Tabla 3: Requisito RU-C-02 51

Tabla 4: Requisito RU-C-03 52

Tabla 5: Requisito RU-C-04 52

Tabla 6: Requisito RU-C-05 53

Tabla 7: Requisito RU-C-06 53

Tabla 8: Requisito RU-C-07 54

Tabla 9: Requisito RU-C-08 54

Tabla 10: Requisito RU-C-09 55

Tabla 11: Requisito RU-C-10 55

Tabla 12: Requisito RU-C-11 56

Tabla 13: Requisito RU-C-12 56

Tabla 14: Requisito RU-C-13 57

Tabla 15: Requisito RU-C-14 57

Tabla 16: Requisito RU-C-15 58

Tabla 17: Requisito RU-C-16 58

Tabla 18: Requisito RU-C-17 59

Tabla 19: Requisito RU-C-18 59

Tabla 20: Requisito RU-C-19 60

Tabla 21: Requisito RU-C-20 60

Tabla 22: Requisito RU-C-21 61

Tabla 23: Requisito RU-C-22 61

Tabla 24: Requisito RU-C-23	62
Tabla 25: Requisito RU-C-24	62
Tabla 26: Requisito RU-C-25	63
Tabla 27: Requisito RU-C-26	63
Tabla 28: Requisito RU-C-27	64
Tabla 29: Requisito RU-C-28	64
Tabla 30: Requisito RU-C-29	65
Tabla 31: Requisito RU-C-30	65
Tabla 32: Requisito RU-C-31	66
Tabla 33: Requisito RU-C-32	66
Tabla 34: Requisito RU-C-33	67
Tabla 35: Requisito RU-C-34	67
Tabla 36: Requisito RU-C-35	68
Tabla 37: Requisito RU-R-01	68
Tabla 38: Requisito RU-R-02	69
Tabla 39: Requisito RU-R-03	69
Tabla 40: Requisito RU-R-04	70
Tabla 41: Requisito RU-R-05	70
Tabla 42: Requisito RU-R-06	71
Tabla 43: Requisito RU-R-07	71
Tabla 44: Requisito RU-R-08	72
Tabla 45: Requisito RU-R-09	72
Tabla 46: Requisito RU-R-10	73
Tabla 47: Requisito RU-R-11	73

Tabla 48: Requisito RU-R-12	74
Tabla 49: Requisito RU-R-13	74
Tabla 50: Requisito RU-R-14	75
Tabla 51: Caso de uso	77
Tabla 52: Caso de uso CU-01	79
Tabla 53: Caso de uso CU-02	80
Tabla 54: Caso de uso CU-03	80
Tabla 55: Caso de uso CU-04	81
Tabla 56: Caso de uso CU-05	82
Tabla 57: Caso de uso CU-06	83
Tabla 58: Caso de uso CU-07	84
Tabla 59: Caso de uso CU-08	85
Tabla 60: Caso de uso CU-09	86
Tabla 61: Caso de uso CU-10	87
Tabla 62: Caso de uso CU-11	88
Tabla 63: Caso de uso CU-12	89
Tabla 64: Caso de uso CU-13	90
Tabla 65: Requisito de software	91
Tabla 66: Requisito RS-F-01	92
Tabla 67: Requisito RS-F-02	93
Tabla 68: Requisito RS-F-03	93
Tabla 69: Requisito RS-F-04	94
Tabla 70: Requisito RS-F-05	94
Tabla 71: Requisito RS-F-06	95

Tabla 72: Requisito RS-F-07	96
Tabla 73: Requisito RS-F-08	96
Tabla 74: Requisito RS-F-09	97
Tabla 75: Requisito RS-F-10	98
Tabla 76: Requisito RS-F-11	99
Tabla 77: Requisito RS-F-12	99
Tabla 78: Requisito RS-F-13	100
Tabla 79: Requisito RS-F-14	100
Tabla 80: Requisito RS-F-15	101
Tabla 81: Requisito RS-F-16	101
Tabla 82: Requisito RS-F-17	102
Tabla 83: Requisito RS-F-18	102
Tabla 84: Requisito RS-F-19	103
Tabla 85: Requisito RS-F-20	103
Tabla 86: Requisito RS-F-21	104
Tabla 87: Requisito RS-F-22	105
Tabla 88: Requisito RS-F-23	106
Tabla 89: Requisito RS-F-24	106
Tabla 90: Requisito RS-F-25	107
Tabla 91: Requisito RS-F-26	108
Tabla 92: Requisito RS-F-27	109
Tabla 93: Requisito RS-F-28	109
Tabla 94: Requisito RS-F-29	110
Tabla 95: Requisito RS-F-30	111

Tabla 96: Requisito RS-F-31	111
Tabla 97: Requisito RS-F-32	112
Tabla 98: Requisito RS-F-33	113
Tabla 99: Requisito RS-F-34	114
Tabla 100: Requisito RS-F-35	114
Tabla 101: Requisito RS-F-36	115
Tabla 102: Requisito RS-F-37	115
Tabla 103: Requisito RS-F-38	116
Tabla 104: Requisito RS-F-39	116
Tabla 105: Requisito RS-F-40	117
Tabla 106: Requisito RS-F-41	117
Tabla 107: Requisito RS-F-42	118
Tabla 108: Requisito RS-N-01	119
Tabla 109: Requisito RS-N-02	119
Tabla 110: Requisito RS-N-03	120
Tabla 111: Requisito RS-N-04	120
Tabla 112: Requisito RS-N-05	121
Tabla 113: Requisito RS-N-06	121
Tabla 114: Requisito RS-N-07	122
Tabla 115: Requisito RS-N-08	122
Tabla 116: Requisito RS-N-09	123
Tabla 117: Matriz de trazabilidad RU-R.....	125
Tabla 118: Prueba de funcionalidad	174
Tabla 119: Prueba PF-01.....	175

Tabla 120: Prueba PF-02.....	176
Tabla 121: Prueba PF-03.....	177
Tabla 122: Prueba PF-04.....	178
Tabla 123: Prueba PF-05.....	180
Tabla 124: Prueba PF-06.....	180
Tabla 125: Prueba PF-07.....	181
Tabla 126: Prueba PF-08.....	182
Tabla 127: Prueba PF-09.....	183
Tabla 128: Prueba PF-10.....	184
Tabla 129: Matriz de trazabilidad RS-PF.....	185
Tabla 130: Planificación de actividades.....	188
Tabla 131: Costes de personal.....	192
Tabla 132: Costes materiales.....	193
Tabla 133: Coste total de desarrollo	193
Tabla 134: Costes del servidor.....	194
Tabla 135: Costes del laboratorio.....	194
Tabla 136: Costes de las herramientas.....	194
Tabla 137: Coste total de despliegue	195

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Línea temporal de Internet [2]	30
Ilustración 2: Crecimiento del número de hosts conectados a Internet [5]	30
Ilustración 3: Principales DBMSs	31
Ilustración 4: Eclipse vs NetBeans	32
Ilustración 5: Logo de Java.....	33
Ilustración 6: JavaScript	34
Ilustración 7: jQuery	35
Ilustración 8: JSON.....	35
Ilustración 9: Objeto en JSON	36
Ilustración 10: Array en JSON	36
Ilustración 11: Valor en JSON	36
Ilustración 12: HTML5.....	37
Ilustración 13: GlassFish	38
Ilustración 14: MySQL.....	39
Ilustración 15: SQLite.....	40
Ilustración 16: Modelo cliente-servidor	41
Ilustración 17: Flujo de las comunicaciones	42
Ilustración 18: Conexión por socket	43
Ilustración 19: Arquitectura basada en WebSockets	44
Ilustración 20: Casos de uso	76
Ilustración 21: Flujo de la aplicación	129
Ilustración 22: Diagrama de componentes [27]	130
Ilustración 23: Base de datos del laboratorio [1]	131

Ilustración 24: Diagrama Entidad-Relación	133
Ilustración 25: Inicio de sesión	134
Ilustración 26: Error en el inicio de sesión	135
Ilustración 27: Reserva en espera	137
Ilustración 28: Eliminación de reserva	137
Ilustración 29: Lista vacía tras la eliminación de reserva	138
Ilustración 30: Reserva lista	138
Ilustración 31: Intento de reserva de fecha pasada	140
Ilustración 32: Confirmación de reserva	141
Ilustración 33: Confirmación de cambio de reserva.....	141
Ilustración 34: Intento de reservar teniendo una reserva comenzada	142
Ilustración 35: Ayuda	143
Ilustración 36: Cuestionario.....	144
Ilustración 37: Confirmación de envío del cuestionario.....	144
Ilustración 38: Configuración.....	145
Ilustración 39: Datos no rellenados completamente	146
Ilustración 40: Fichero de datos no elegido	146
Ilustración 41: Fichero no válido	147
Ilustración 42: Página de espera	148
Ilustración 43: Estado inicial de la página de ejecución (pantallazo)	149
Ilustración 44: Ejecución (pantallazo).....	150
Ilustración 45: Estado inicial de la página de ejecución (foto)	150
Ilustración 46: Ejecución (foto).....	151
Ilustración 47: Opción de guardado	151

Ilustración 48: Opción de abrir o guardar el fichero	152
Ilustración 49: Fichero creado	152
Ilustración 50: Contenido del fichero	153
Ilustración 51: Diagrama de navegación	154
Ilustración 52: Diagrama de despliegue [27]	171
Ilustración 53: Diagrama de Gantt.	189
Ilustración 54: Guía de usuario - Inicio de sesión.....	204
Ilustración 55: Guía de usuario - Ayuda	205
Ilustración 56: Guía de usuario - Cuestionario	206
Ilustración 57: Guía de usuario - Reserva de herramientas	208
Ilustración 58: Guía de usuario - Estados de reserva	209
Ilustración 59: Guía de usuario - Configuración	210
Ilustración 60: Guía de usuario - Página de espera	211
Ilustración 61: Guía de usuario - Ejecución	212
Ilustración 62: Guía de usuario - Guardado de datos.....	214

1. Introducción

En este capítulo se ofrecerá una visión general sobre el proyecto y sus objetivos. Se detallarán las fases de desarrollo por las que se ha transitado y el contenido del presente documento.

1.1. Visión general

Este proyecto implementa una aplicación web que permite a los alumnos de una determinada institución (por ejemplo: UC3M) acceder a recursos hardware que se encuentran en una ubicación remota e interactuar con éstos.

El proyecto está basado en otro proyecto anterior: “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha. En dicho proyecto se creaban los conceptos “laboratorio” y “herramienta”. Un laboratorio era un ordenador que ofrecía la posibilidad de usar herramientas que en él estuviesen registradas [1].

Este proyecto utiliza la infraestructura de comunicaciones creada en el proyecto anterior, a la cual se añade un front-end para la realización de prácticas para el control remoto de un motor, configurando un controlador de tipo PID. Este proyecto incluye toda la gestión de la práctica y la parte del front-end, mientras que el middleware de comunicaciones está basado en el proyecto referenciado anteriormente (con algunas modificaciones) y la parte del control del motor y el motor mismo han sido simulados.

Un estudiante, previamente dado de alta en la base de datos de la aplicación por el administrador de ésta, podrá acceder a la aplicación a través de su nombre de usuario y contraseña y utilizar los servicios ofrecidos. La aplicación tiene varias funcionalidades auxiliares: brindar temas de ayuda al usuario y permitir el rellenado y envío de un cuestionario con fines estadísticos sobre la práctica realizada. Sin embargo, el servicio principal es la ejecución de la práctica. Todos los servicios ofrecidos por la aplicación serán gestionados por un servidor, tanto el envío y recuperación de datos, como la comunicación con los laboratorios.

Una práctica es, a efectos de la aplicación, una reserva realizada por un usuario sobre un conjunto de herramientas que se perciben como una sola entidad. La aplicación permite al usuario reservar ese conjunto de herramientas por un tiempo determinado (predefinido), garantizando la exclusividad de uso de dicho usuario sobre el conjunto durante ese tiempo. El usuario puede elegir cuándo reservar y, si así lo desea, cancelar y cambiar la fecha de su reserva que, para el sistema implementado, será única

(aunque se puede reservar de nuevo). En base a la fecha elegida por el usuario para efectuar la práctica, éste podrá comenzar la ejecución.

El control del motor se realiza en 2 pasos. Primeramente, se configura el controlador PID del motor mediante sus parámetros de configuración P (Proporcional), I (Integral) y D (Derivativo). Después se genera una señal de entrada desde el front-end (configurada por el alumno). Esta señal se implementa de forma discreta mediante el envío periódico del valor actual de la señal. Esta señal será recibida por el controlador PID que, a su vez, estará conectado al motor mediante un bucle de control retroalimentado. Por último la señal de salida del motor es reenviada al front-end de forma discreta mediante una serie de valores periódicos.

La aplicación ofrece una interfaz que contiene el video en tiempo real del motor y dibuja en una gráfica las señales de entrada y salida. Adicionalmente, permite el guardado de los datos resultantes de la ejecución para su posterior análisis por el usuario.

Este es, en términos generales, el funcionamiento del proyecto.

1.2. Motivación

El constante desarrollo tecnológico al que se asiste en los tiempos actuales, estimula la búsqueda constante de soluciones cada vez más orientadas a la comodidad de los usuarios finales de los servicios.

En el entorno universitario, en el que el estudiante debe organizar su tiempo cuidadosamente, la idea de poder elegir cuándo realizar una práctica y desde dónde realizarla, se vuelve más necesidad que idea.

Desde el punto de vista del estudiante, la posibilidad de reservar y acceder a un recurso ubicado remotamente, aporta las siguientes ventajas:

- El estudiante elige en qué momento realizar la práctica.
- El estudiante puede hacer la práctica desde cualquier ordenador con los requisitos mínimos que pide la aplicación, evitándose desplazamientos.

Esta aplicación busca la comodidad del usuario que, desde cualquier ordenador con Internet y un navegador con requerimientos mínimos, puede acceder a un recurso localizado en una ubicación remota.

1.3. Objetivos

El objetivo de este proyecto es el desarrollo de una aplicación web intuitiva y de fácil uso que permita la reserva, acceso e interacción con dos herramientas específicas de un laboratorio, como parte de una práctica de una determinada asignatura.

Como objetivos más específicos y principales, se plantean los siguientes:

- Mostrar video en tiempo real de un motor ubicado en un laboratorio remoto.
- Enviar una señal de entrada definida por el usuario a un sistema de control retroalimentado compuesto de un controlador PID y un motor. Así mismo, recibir la señal de salida correspondiente y dibujar una gráfica con el par asociado.

Para el cumplimiento de dichos objetivos, además de la aplicación web, se han implementado dos herramientas en un mismo laboratorio:

- **RLF_Cam:** activa la cámara ubicada en el laboratorio remoto, que grabará el comportamiento del motor.
- **RLF_Chart:** simula el comportamiento de un sistema de control retroalimentado compuesto por un controlador PID y un motor.

Por otro lado, para mantener un control del uso de las herramientas y evitar conflictos y saturación en la ejecución de éstas, se implementará la aplicación de modo que el usuario solo pueda acceder a las herramientas si las ha reservado previamente.

Como objetivo secundario se encuentra la recopilación de opiniones de los usuarios sobre la práctica realizada, a través de un cuestionario.

A pesar de que esta aplicación está dirigida a una práctica específica que utiliza solo dos herramientas, se pretende que ésta sea escalable, que permita gestionar varias prácticas y herramientas en un futuro, con modificaciones mínimas.

1.4. Fases de desarrollo

Las fases por las que ha transitado el desarrollo de este proyecto son las siguientes:

- **Propuesta:** reunión con el cliente, para el planteamiento en términos generales de los objetivos y funcionalidades del proyecto.
- **Análisis:** investigación de las tecnologías y arquitecturas existentes y recomendables para el tipo de sistema que se implantará, y elección de las que se usarán en el proyecto. Captura y análisis de los requisitos de la aplicación que se desarrollará, a través de reuniones con el cliente.
- **Diseño:** diseño de la arquitectura del sistema, tomando como base los requisitos establecidos, derivados de la fase de análisis. Todos los requisitos se tomarán en cuenta y se garantizará la inclusión de cada uno de ellos en el diseño.
- **Implementación:** implementación, a través de código, de las funcionalidades de la aplicación, siguiendo las pautas marcadas por el diseño.
- **Pruebas:** verificaciones exhaustivas de las funcionalidades del sistema con la finalidad de encontrar errores y/o características mejorables, y darles solución y/o implementar las mejoras, respectivamente.
- **Documentación:** documentación del trabajo realizado durante el desarrollo del proyecto.

1.5. Contenido del documento

Este documento se dividirá, para su desarrollo, en los siguientes capítulos:

- **Capítulo 1: Introducción**
Proporciona una visión general del proyecto, los objetivos que se plantean para éste y la motivación para su desarrollo.
- **Capítulo 2: Estado del arte**
Realiza un repaso por la evolución de tecnologías claves para el desarrollo de este proyecto. Especifica las tecnologías utilizadas para la implementación, justificando cada elección.
- **Capítulo 3: Análisis de la solución**

Especifica las funcionalidades y restricciones de la aplicación, planteadas a través de los requisitos concretos de ésta.

➤ **Capítulo 4: Diseño**

Define la arquitectura del sistema, de los datos y las interfaces que proporcionará la aplicación. Identifica los componentes de la aplicación y la relación entre éstos.

➤ **Capítulo 5: Implementación**

Detalla los métodos de comunicación cliente-servidor y el funcionamiento de las herramientas creadas.

➤ **Capítulo 6: Pruebas**

Define las pruebas que se han realizado sobre el sistema para garantizar su correcto funcionamiento y los resultados arrojados por dichas pruebas.

➤ **Capítulo 7: Planificación**

Muestra la planificación que se ha seguido durante el desarrollo del proyecto, indicando el tiempo utilizado para cada fase de éste.

➤ **Capítulo 8: Presupuesto**

Especifica los gastos generados durante el desarrollo del sistema y los que se derivarían del despliegue del mismo.

➤ **Capítulo 9: Conclusiones**

Indica las conclusiones a las que se han llegado tras la realización del proyecto. Define las optimizaciones y adiciones que se podrían llevar a cabo sobre el sistema en un futuro próximo.

Adicionalmente, se incluyen los siguientes anexos:

➤ **Anexo I: Guía de usuario**

Indica al usuario final de la aplicación cómo utilizar los servicios de ésta.

➤ **Anexo II. Guía de despliegue**

Indica los pasos a seguir para la instalación y correcta configuración de la aplicación.

2. Estado del arte

Con el objetivo de comprender las tecnologías elegidas para el desarrollo de este proyecto, en este capítulo se analizará la evolución de dichas tecnologías. Seguidamente, se especificarán las herramientas, lenguajes, modelos de trabajo y recursos utilizados.

2.1. Evolución de la tecnología

Internet, desde sus inicios, no ha hecho más que expandirse y evolucionar con el paso del tiempo y la inclusión de nuevos avances tecnológicos. Su enorme éxito comercial, sumado a la aparición de las bases de datos relacionales, creó un nuevo tipo de aplicaciones en las que se fusionan ambas tecnologías. A continuación, se hará un recorrido por la historia de Internet y las bases de datos.

2.1.1. Comunicaciones

En la historia de las comunicaciones, nada ha sido más revolucionario que el nacimiento de Internet, el medio por excelencia para la transmisión de información a nivel mundial y la interacción entre individuos/ordenadores sin importar su localización geográfica.

El primer concepto similar al de Internet, lo brindó J.C.R. Licklider, jefe del programa de investigación de computadores en DARPA, en agosto de 1962, en una serie de memorándums. En ellos se discutía sobre la idea de un conjunto de computadores interconectados globalmente, a través de los cuales cualquier persona podría acceder a datos y programas. En julio de 1961, Leonard Kleinrock de MIT publicaba el primer artículo sobre teoría de conmutación de paquetes y, en 1964, el primer libro sobre este tema. En 1965 se lograba conectar el computador TX-2 en Mass al computador Q-32 en California a través de una línea telefónica, creando la primera WAN de la historia. En 1967, Lawrence G. Roberts publicaba su plan para ARPANET y, en octubre del 1972, tras varios cambios, adiciones y optimizaciones en la estructura y especificación, y el trabajo conjunto de varios investigadores, se realizaba la primera demostración pública de ARPANET. En 1972 también, se creaba el software de envío y lectura de correo electrónico. En julio se añadían las funcionalidades de listar, guardar, reenviar y responder a los mensajes. Con el tiempo, ARPANET fue creciendo y se convirtió en lo que hoy se conoce como Internet. Más adelante, se creaba el protocolo TCP/IP [2] [3].

El desarrollo de LANs, PCs y estaciones de trabajo en los años 80, permitió el florecimiento de Internet y, el consecuente crecimiento del número de redes, promovió la aparición del DNS.

En octubre de 1995, el Federal Networking Council definió el término Internet como un “sistema global de información que [4]:

- está relacionado lógicamente por un único espacio de direcciones global basado en el protocolo de Internet (IP) o en sus extensiones.
- es capaz de soportar comunicaciones usando el conjunto de protocolos TCP/IP o sus extensiones u otros protocolos compatibles con IP.
- emplea, provee, o hace accesible, privada o públicamente, servicios de alto nivel en capas de comunicaciones y otras infraestructuras relacionadas aquí descritas.”

Internet fue concebido en la era de la compartición de tiempo, pero ha sobrevivido en la era de los ordenadores personales, las arquitecturas cliente-servidor y peer-to-peer, las LANs y el ATM. Ha evolucionado para brindar servicios como la compartición de ficheros y recursos, el acceso remoto, el correo electrónico y se ha convertido en un éxito comercial con billones de euros de inversión anual. Esta continua evolución, más la posibilidad de casi todos de tener ordenadores personales, móviles, etc., y las crecientes exigencias de los usuarios, sugiere que los cambios y optimizaciones en Internet seguirán sucediéndose en aras de proveer servicios más especializados y enfrentar los retos que el desarrollo tecnológico plantea.

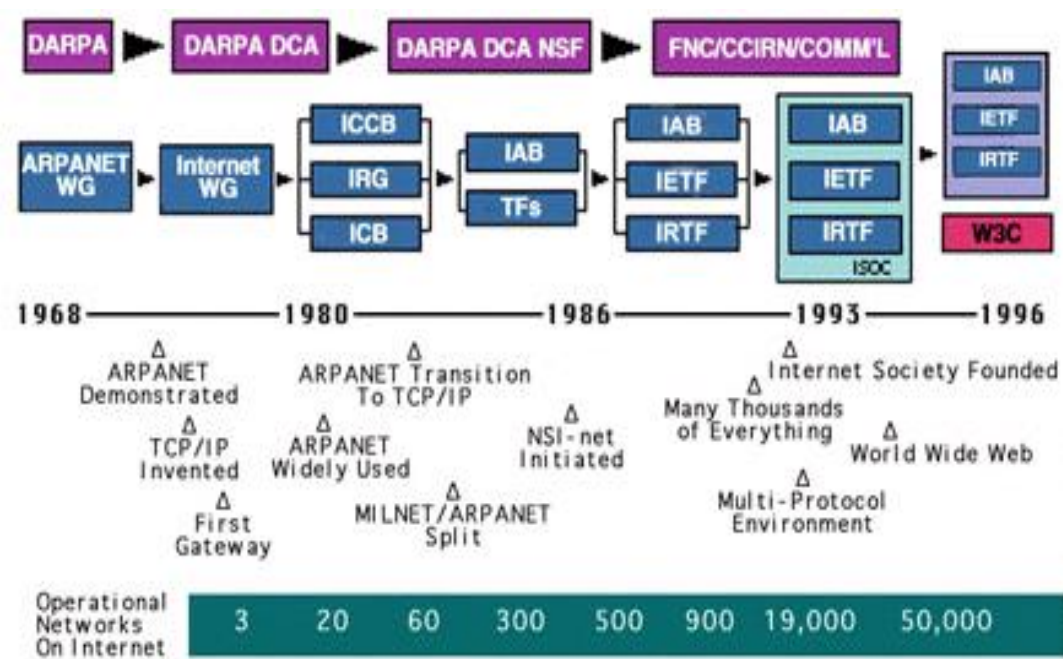


Ilustración 1: Línea temporal de Internet [2]

(imagen tomada de <http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet#History>)

Internet Domain Survey Host Count

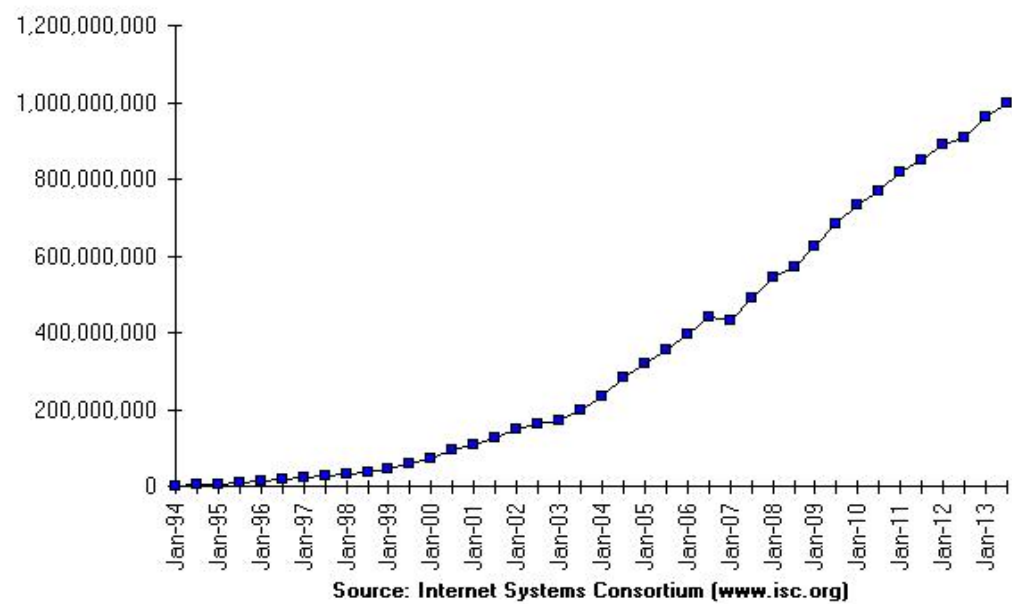


Ilustración 2: Crecimiento del número de hosts conectados a Internet [5]

(imagen tomada de <https://www.isc.org/services/survey/>)

2.1.2. Almacenamiento

Una base de datos es una colección de información y medios para manipular datos, que provee almacenamiento para grandes cantidades de datos y acceso rápido a estos [7]. Un sistema de gestión de bases de datos es un conjunto de programas que definen, guardan, manipulan y controlan los datos en una base de datos. Aunque el almacenamiento de información existe desde mucho antes, las primeras bases de datos computarizadas surgieron en los años 60, cuando el uso de los ordenadores se convirtió en una opción viable para las organizaciones privadas.

Entre 1968 y 1980 se desarrollaron las primeras bases de datos no relaciones y las primeras bases de datos relaciones surgieron en los 70s. Con la evolución de las bases de datos relacionales, surgió la necesidad de un lenguaje de consultas estandarizado y, de este modo, en los 80s, SQL se convirtió en el lenguaje estándar. Las bases de datos relacionales se convirtieron rápidamente en un éxito y, a mediados de los 90s, al incorporarse la orientación a objetos a las bases de datos relacionales, surgieron nuevas áreas de aplicación y las primeras aplicaciones de bases de datos para Internet [6]. En la actualidad, con el desarrollo de Internet, la tecnología de bases de datos continúa evolucionando. Las principales compañías que desarrollan esta tecnología son: Microsoft, IBM and Oracle.



Ilustración 3: Principales DBMSs

(imágenes tomadas de http://www.tecnologianirvana.com/services/images/mysql_logo.png,
http://www.tecnologianirvana.com/services/images/oracle_db.png y
<http://www.tecnologianirvana.com/services/images/sqlserver.png>)

2.2. Tecnología utilizada

Para el desarrollo de este proyecto se han utilizado diversas herramientas. Seguidamente se describirán estas herramientas y se explicarán sus funciones en la aplicación.

2.2.1. Entorno de desarrollo

Existen varios entornos de desarrollo de aplicaciones web con Java. Entre ellos se puede citar a JDeveloper, Eclipse, JBuilder, IntelliJ IDEA y NetBeans IDE. Se pueden destacar como los más populares Eclipse y NetBeans.



Ilustración 4: Eclipse vs NetBeans

(imagen tomada de http://3.bp.blogspot.com/-UOvTJydHV5M/Uloj_Gg49zI/AAAAAAAAABrk/UqLs-O82YFw/s320/Eclipse+vs+Netbeans.bmp)

Ambos entornos permiten el desarrollo de aplicaciones de escritorio, móviles y web en Java, así como aplicaciones en HTML5, utilizando HTML, JavaScript y CSS.

En la elección entre estos dos entornos de desarrollo se tomaron en cuenta varios factores [8] [9]:

- **Sistemas operativos:** ambos entornos funcionan en la mayoría de plataformas, siempre y cuando esté instalada la máquina virtual de Java.
- **Lenguajes de programación:** ambos entornos proporcionan mecanismos para una amplia variedad de lenguajes, incluyendo los necesarios para este proyecto. Sin embargo, en Eclipse, el soporte para muchos de los lenguajes viene a través de plugins que hay que añadir.
- **Bases de datos:** NetBeans tiene soporte integrado para Oracle, SQL y MySQL. Eclipse tiene soporte para JDBC, pero configurar la conexión es bastante más complicado.
- **Facilidad de uso:** NetBeans es más amigable e intuitivo.

- **Disponibilidad:** ambos son gratuitos.
- **Servicios integrados:** muchas de las funcionalidades de Eclipse tienen que añadirse a través de plugins. NetBeans tiene integradas todas las funcionalidades necesarias para este proyecto.

La versión *netbeans-7.3.1-linux* traía soporte integrado para GlassFish, HTML5 y WebSockets, elementos esenciales para la aplicación. En Eclipse, sin embargo, se tendrían que haber buscado e instalado todos los plugins necesarios. En términos de tiempo y comodidad, NetBeans era la mejor elección, por lo que se ha optado por usar este IDE para el proyecto.

Se usará la versión *netbeans-7.3.1-linux* y con esta herramienta se crearán tanto el cliente como el servidor que se comunicará con los laboratorios.

2.2.2. Lenguajes de desarrollo

2.2.2.1. Java

Java es un lenguaje de programación y una plataforma que se usa para el desarrollo de aplicaciones. Es la base de prácticamente todos los tipos de aplicaciones de red y el estándar global para el desarrollo y distribución de juegos, contenido web, software empresarial y aplicaciones móviles [10].



Ilustración 5: Logo de Java

(imagen tomada de <http://www.java.com/im/javalogo-81x162.png>)

Los siguientes datos han sido extraídos de la página oficial de Java:

- “El 97% de los escritorios empresariales ejecutan Java.
- El 89% de los escritorios (o computadoras) en Estados Unidos ejecutan Java.

- Existen 9 millones de desarrolladores de Java en todo el mundo.
- Java es la primera opción para los desarrolladores.
- Java es la primera plataforma de desarrollo a nivel mundial.
- 3 mil millones de teléfonos móviles ejecutan Java.
- El 100% de los reproductores de Blu-ray incluyen Java.
- Existen 5 mil millones de Java Cards en uso.
- 125 millones de dispositivos de televisión ejecutan Java.
- 5 de los 5 principales fabricantes de equipos originales utilizan Java ME.”

Para la implementación de la parte del servidor de esta aplicación se utilizará dicho lenguaje. Esta decisión se debe a que el código de los laboratorios remotos está escrito en Java. Desde el punto de vista de facilitar las comunicaciones entre el servidor y los laboratorios a través de sockets, Java es la mejor opción.

2.2.2.2. JavaScript

JavaScript es un lenguaje ligero e interpretado, orientado a objetos, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador. Es utilizado en millones de páginas web y aplicaciones en el mundo [11].



Ilustración 6: JavaScript

(imagen tomada de <http://www.agencia-internet.es/wp-content/uploads/2012/10/JavaScriptLogo.png>)

Este lenguaje será utilizado para el desarrollo de la parte cliente de esta aplicación.

➤ **jQuery**

jQuery es una librería de JavaScript que simplifica la manipulación de documentos HTML, el manejo de eventos, el desarrollo de animaciones y funciona en multitud de navegadores. Su versatilidad, rapidez, amplitud de recursos y extensibilidad ha cambiado el modo de escribir código JavaScript [12].



Ilustración 7: jQuery

(imagen tomada de <http://brand.jquery.org/resources/jquery-mark-dark.gif>)

jQuery se ha utilizado en la parte del cliente, a través de varios plugins que se han añadido a la aplicación web para un mayor impacto visual y/o la simplificación de varias funcionalidades.

➤ **JSON**

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos, basado en un subconjunto de JavaScript [13].



Ilustración 8: JSON

(imagen tomada de <http://www.json.org/img/json160.gif>)

JSON utiliza dos estructuras: una colección de pares de nombre/valor y una lista ordenada de valores.

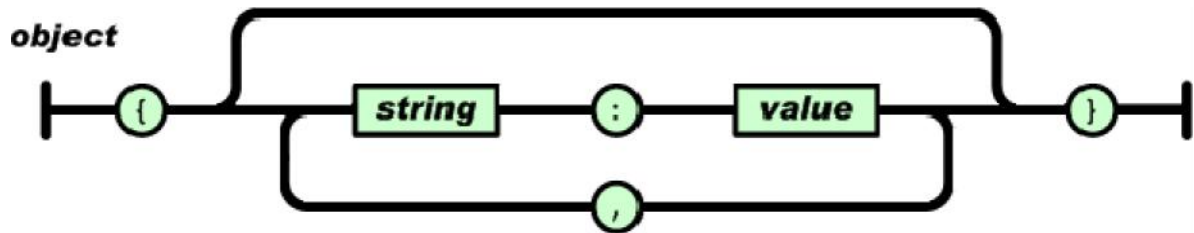


Ilustración 9: Objeto en JSON

(imagen tomada de <http://www.json.org/object.gif>)

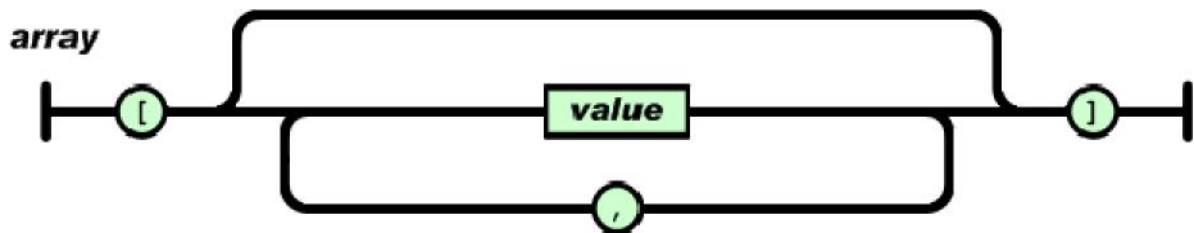


Ilustración 10: Array en JSON

(imagen tomada de <http://www.json.org/array.gif>)

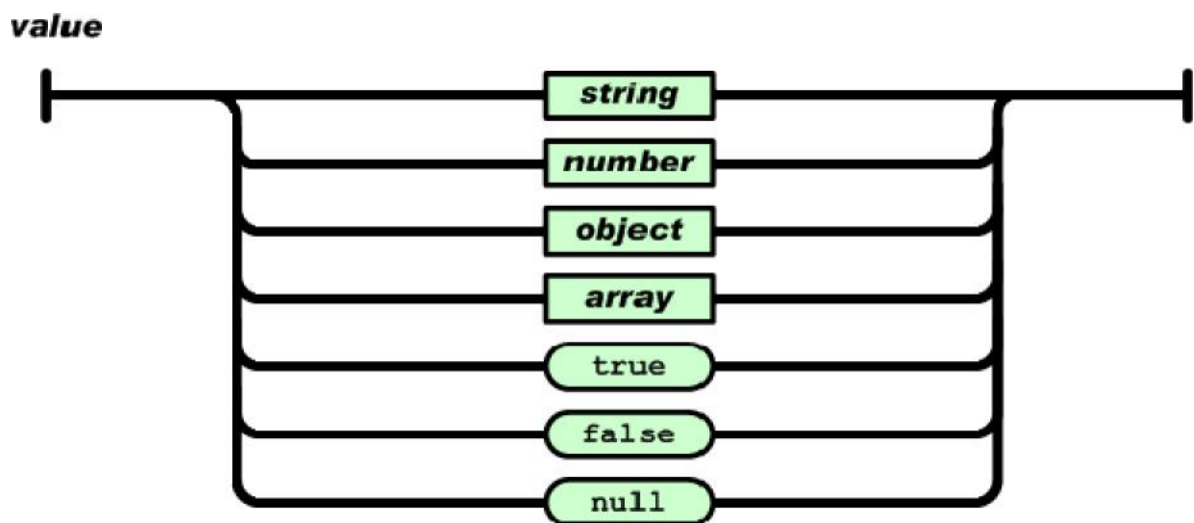


Ilustración 11: Valor en JSON

(imagen tomada de <http://www.json.org/value.gif>)

JSON se utilizará para enviar mensajes del cliente al servidor y viceversa. En el lado del cliente, el mensaje enviado al servidor a través del websocket será un JSON conteniendo la acción a ejecutar por el servidor y los datos que éste necesita para ejecutarla. En el lado del servidor, tras ejecutar la acción indicada,

la respuesta enviada al cliente será un JSON con los datos que éste había solicitado. Así mismo, el servidor enviará JSONs con datos destinados al cliente, sin petición previa de éste, durante la ejecución de las herramientas.

2.2.2.3. HTML

HTML (HyperText Markup Language) es un lenguaje de marcado para la creación de páginas web. Se utiliza para definir la estructura y semántica del contenido de un documento web [14]. Las especificaciones de HTML son reguladas por el World Wide Web Consortium.

El nuevo estándar HTML es HTML5. HTML5 añade nuevos elementos y atributos, facilidades para el streaming de audio y video y la creación de gráficos, e introduce el concepto Web Storage [15].



Ilustración 12: HTML5

(imagen tomada de https://developer.mozilla.org/files/3563/HTML5_Logo_128.png)

En esta aplicación se usa HTML5 para definir las páginas web. HTML solamente describe el contenido y su significado. Del estilo y el formato del contenido se encargan las CSS.

2.2.2.4. CSS

Las CSS (Cascading Style Sheets) son un mecanismo para añadir estilo (fuentes, colores, espaciado, etc.) a los documentos web [16] [17]. **CSS** es una especificación del W3C.

2.2.2.5. Shell script

Los shell scripts son archivos de texto que contienen una secuencia de comandos que son ejecutados por bash [18]. En este proyecto, se han utilizado shell scripts para definir las funcionalidades de las nuevas herramientas que han sido creadas. Estas herramientas han sido definidas según las especificaciones del proyecto “Diseño y

desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha.

2.2.3. Servidor de aplicaciones

Un servidor de aplicaciones es un tipo de servidor diseñado para instalar, operar y albergar aplicaciones y servicios para los usuarios finales y organizaciones. Facilita el acceso a aplicaciones que son usadas simultáneamente por múltiples usuarios conectados local o remotamente. En este proyecto se ha utilizado GlassFish.

GlassFish Server Open Source Edition es un servidor de aplicaciones compatible con Java EE, desarrollado por Oracle y la comunidad Glassfish bajo una licencia de código abierto [19]. GlassFish normalmente es muy rápido en actualizarse y añadir soporte para las nuevas versiones de Java, una vez estas son publicadas.



Ilustración 13: GlassFish

(imagen tomada de https://glassfish.java.net/glassfish_buttons/glassfish_logo.gif)

Existen otros servidores de aplicaciones populares que se podrían haber elegido para el desarrollo del proyecto, como WebSphere y JBoss. Sin embargo, el primero no es de software libre y el segundo no tiene soporte para WebSockets. Esto lleva a elegir GlassFish, un servidor rico en funcionalidad, de código abierto y con soporte para los recursos que se van a utilizar en la aplicación.

2.2.4. Almacenamiento

2.2.4.1. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario de código abierto [20]. Es uno de los DBMS más populares por su velocidad, fiabilidad y facilidad de uso.



Ilustración 14: MySQL

(imagen tomada de http://www.tecnologianirvana.com/services/images/mysql_logo.png)

En este proyecto se ha utilizado MySQL para el manejo de los datos que utilizará la aplicación web (usuarios, reservas, respuestas al cuestionario, etc). Así mismo, se utilizarán las herramientas `mysql-query-browser` y `mysql-admin`, según las especificaciones del proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha.

Se ha elegido MySQL por las características mencionadas anteriormente y por comodidad, al haberse usado MySQL para las bases de los laboratorios remotos. Por otro lado, NetBeans, el entorno de desarrollo elegido, tiene soporte integrado para MySQL.

Entre las posibles elecciones, además de MySQL, se encontraban Oracle Database y Microsoft SQL Server. Los tres DBMS tienen un alto rendimiento, amplias capacidades y un alto nivel de seguridad. Sin embargo, Oracle Database y Microsoft SQL Server no son software libre y Microsoft SQL Server necesita instalación de drivers y varias modificaciones para ejecutarse en Linux, dado que no es un software multiplataforma. Por lo tanto, se ha elegido MySQL, DBMS libre, gratuito, potente y con las capacidades necesarias para el desarrollo del proyecto.

2.2.4.2. SQLite

SQLite es una librería que implementa un motor de base de datos SQL transaccional, auto contenido y sin servidor [21]. Su código es público y libre de utilizarse para cualquier propósito, tanto privado como comercial. Su tamaño es pequeño en disco y responde bien ante errores de disco, fallos de memoria y problemas de energía.



Ilustración 15: SQLite

(imagen tomada de http://www.sqlite.org/images/sqlite370_banner.gif)

En este proyecto se ha utilizado SQLite como parte de las especificaciones para la creación de herramientas detalladas en el proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha.

2.2.5. Modelo de trabajo

La arquitectura cliente/servidor es un modelo de aplicación distribuida en el cual la máquina que hace de servidor administra los recursos a ser utilizados por el cliente. En esta arquitectura, uno o más máquinas cliente están conectadas a un servidor a través de una red o conexión a Internet. El servidor alberga y provee servicios a los clientes, bajo la demanda de estos, tales como: acceso a aplicaciones, almacenamiento, compartición de archivos, acceso a hardware, etc. Los clientes envían peticiones al servidor y éste, tras procesarlas, envía las respuestas/acciones correspondientes solicitadas por los clientes. El servidor atiende múltiples conexiones y peticiones de clientes simultáneamente y se mantiene en espera de dichas peticiones. Típicamente, el servidor y el cliente residen en máquinas distintas, pero pueden ejecutarse en la misma.

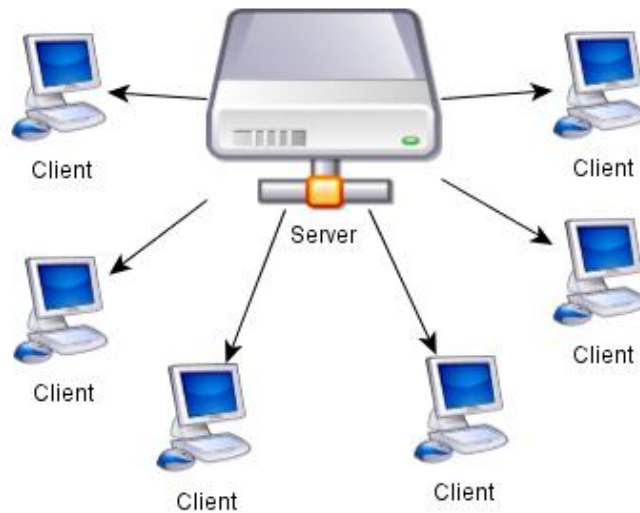


Ilustración 16: Modelo cliente-servidor

(imagen tomada de <http://www.vorlesungen.info/sites/default/files/Client-Server-Modell.jpg>)

En este proyecto se utilizará este modelo. El funcionamiento, brevemente descrito, será el siguiente:

- La aplicación cliente envía peticiones al servidor.
- El servidor, en dependencia del tipo de petición, hace consultas a la base de datos o se conecta a laboratorios remotos.
 - i) El servidor recibe las respuestas a las consultas y las devuelve a la aplicación cliente (si fuese necesario).
 - ii) Se establece la conexión entre servidor y laboratorio y comienza el uso de las herramientas, cuyas respuestas (si las hubiese) se envían al cliente a través del servidor. Una vez que, a petición del cliente, el servidor haya iniciado la herramienta, comenzará a enviar datos al cliente sin mediar ninguna petición y sin afectar la ejecución de éste (comunicación asíncrona).

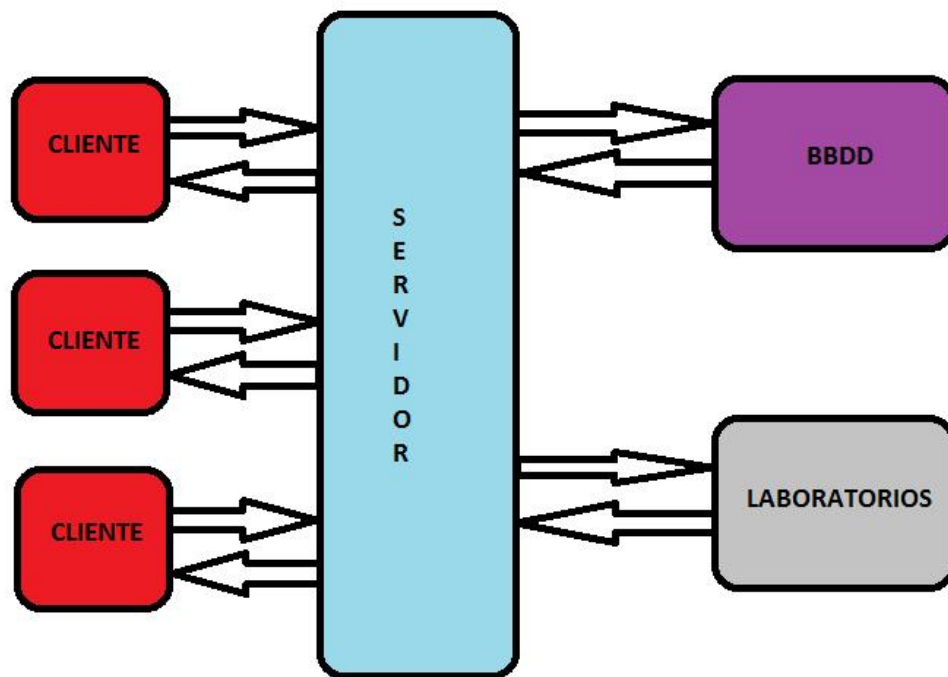


Ilustración 17: Flujo de las comunicaciones

2.2.6. Comunicaciones

2.2.6.1. Sockets

Un socket es un extremo de un link de comunicación bidireccional entre dos programas que corren en una red. Un socket está asociado a un número de puerto, de modo que la capa TCP pueda identificar a qué aplicación se deben enviar los datos [22]. Un extremo es una combinación de una dirección IP y un número de puerto, y cada conexión TCP puede ser identificada de modo unívoco por dos extremos.

En la comunicación por sockets, el servidor tiene un socket asociado a un número de puerto específico. El servidor simplemente espera a que el cliente envíe por el socket una petición de conexión. En el lado del cliente, éste conoce el nombre de la máquina en la que corre el servidor y el número de puerto en que el servidor está escuchando. El cliente realiza una petición de conexión y, si el servidor la acepta, se crea en el servidor un nuevo socket asociado al mismo puerto de escucha, pero estando el otro

extremo de la comunicación asociado a la IP y puerto del cliente. De este modo, el servidor tiene un socket por el que continúa escuchando peticiones de conexión, y otro socket para atender las peticiones del cliente conectado. El cliente y el servidor se comunican a través de este nuevo socket en el que ambos pueden leer/escribir.

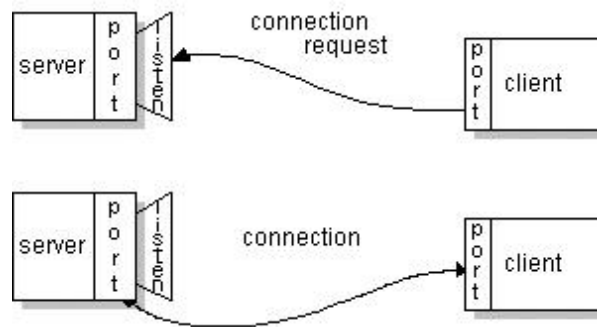


Ilustración 18: Conexión por socket

(imágenes tomadas de <http://docs.oracle.com/javase/tutorial/figures/networking/5connect.gif> y <http://docs.oracle.com/javase/tutorial/figures/networking/6connect.gif>)

Los sockets permiten a los programas utilizar los comandos del sistema operativo para el manejo de funciones de red, lo cual facilita el trabajo de los desarrolladores, que no tendrían que construir desde cero las conexiones de red para cada programa [23].

En este proyecto se han utilizado sockets para la comunicación entre el servidor y los laboratorios/herramientas. Este modo de comunicación se ha elegido debido a que el software para los laboratorios remotos (implementado en otro proyecto) utiliza sockets para la escucha de peticiones de conexión y recepción/envío de datos a las herramientas.

➤ **WebSockets**

La especificación WebSocket, desarrollada como parte de la iniciativa HTML5, brinda la interfaz WebSocket JavaScript, que define una conexión de socket sobre la cual se pueden enviar mensajes entre cliente y servidor [24]. Este nuevo estándar simplifica la complejidad de las comunicaciones web bidireccionales y el manejo de conexiones [25].

La especificación HTML5 WebSockets define una API que permite a las páginas web usar el protocolo WebSocket para una comunicación bidireccional con un host remoto. El siguiente diagrama muestra una arquitectura básica basada en WebSockets en la cual los navegadores usan una conexión WebSocket para una comunicación full-duplex y directa con un host remoto.

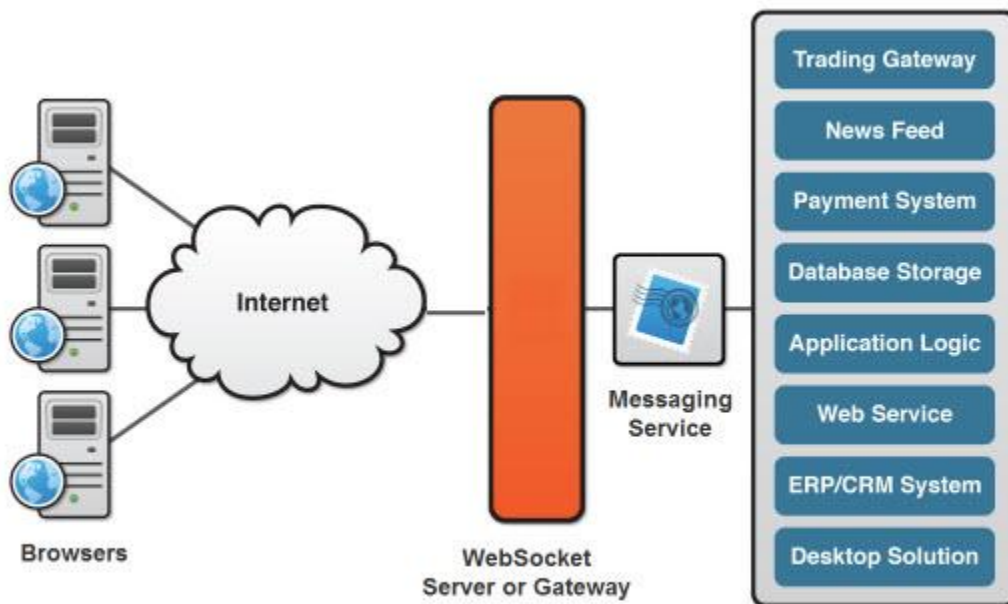


Ilustración 19: Arquitectura basada en WebSockets

(imagen tomada de <http://www.websocket.org/img/websocket-architecture.jpg>)

Los WebSockets son una tecnología avanzada que hace posible el establecimiento de una sesión de comunicación interactiva entre el navegador del usuario y el servidor. Con esta API, se pueden enviar mensajes al servidor y recibir respuestas, a través de eventos, sin tener que consultar continuamente al servidor.

Las tecnologías Comet, push y long-polling son demasiado lentas para aplicaciones web, dado que normalmente el servidor se bloquea tras una petición hasta que se devuelva una respuesta al cliente. Los WebSockets ofrecen una verdadera capacidad de comunicación bidireccional entre cliente y servidor que el resto de tecnologías no tienen (intentan emular esta capacidad,

pero ineficientemente, dado la enorme sobrecarga y el gasto de recursos), ni siquiera SOAP o REST [26]. Comparado con REST, Websockets se destaca por ser más ligero y reducir la latencia. Una vez establecida una sesión, los mensajes pueden tener una sobrecarga por mensaje de solo 2 bytes. REST, sin embargo, añade cabeceras HTTP relativamente grandes en cada petición y respuesta.

Con WebSockets, el servidor puede enviar información al cliente en cualquier momento, simultáneamente al cliente enviándole peticiones, sin la dependencia de una petición previa por parte del cliente. A diferencia de REST, la comunicación a través de WebSockets es típicamente asíncrona (el cliente no espera por la respuesta del servidor, sino que sigue con su ejecución tras realizar la petición a éste).

Para este proyecto, se he elegido WebSockets, por la necesidad de una comunicación asíncrona con el cliente. El cliente tendrá también comportamiento síncrono en algunos casos, en los que simplemente hará peticiones al servidor y esperará por respuestas. En otros casos, el servidor enviará datos al cliente periódicamente, sin haber mediado una petición previa del cliente.

3. Análisis de la solución

En este capítulo se analizarán los requisitos y especificaciones que debe seguir la aplicación para cumplir las expectativas del cliente, y se describirán detalladamente todas sus funcionalidades.

3.1. Descripción de la solución

3.1.1. Capacidades generales

La aplicación web RLF está centrada en dos herramientas específicas:

- **RLF_Cam:** Accede a la cámara de un ordenador (laboratorio) remoto y transmite el video de esta cámara a una dirección IP y un puerto especificados.
- **RLF_Chart:** Recibe los valores P, I y D, y luego recibe, uno a uno, un listado de números enviados por el usuario. Los pasa por el PID y devuelve las salidas correspondientes.

Teniendo en cuenta estas herramientas, la aplicación:

- Provee de un mecanismo para que los usuarios puedan conectarse y utilizar de modo remoto, a través de un navegador, ciertos recursos.
- Ofrece a los usuarios registrados en la aplicación la posibilidad de reservar herramientas ubicadas en laboratorios remotos por un determinado tiempo.
- Permite a los usuarios modificar la fecha de las reservas o eliminarlas, siempre y cuando las reservas no hayan comenzado.
- Muestra a los usuarios sus listas de reservas y actualiza estas listas en base a la adición/finalización de reservas.
- Permite el uso exclusivo por el usuario de las herramientas reservadas y el almacenamiento de los resultados obtenidos durante la ejecución de éstas si el usuario así lo desea.
 - Recibe y reproduce el video enviado por el laboratorio remoto en tiempo real.
 - Envía periódicamente datos a las herramientas, recibe las respuestas de éstas y las dibuja en una gráfica.
- Ofrece la posibilidad de rellenar un cuestionario sobre la práctica realizada.
- Ofrece temas de ayuda al usuario para un uso óptimo de la aplicación.

3.1.2. Restricciones generales

El diseño de la aplicación web RLF cumple las siguientes restricciones:

- La aplicación será independiente del Sistema Operativo en el que se ejecute.
- La aplicación funcionará en navegadores que soporten WebSockets.
- La aplicación solamente podrá ser usada por usuarios que figuren en la base de datos de ésta. No se ofrecerá el servicio de registro de usuarios.

3.1.3. Características de los usuarios

En este proyecto se identifica un solo tipo de usuario:

- **Usuario registrado en la aplicación:** típicamente, un estudiante de ingeniería que podrá acceder de modo remoto a laboratorios de la Universidad y las herramientas que éstos alberguen. Evidentemente, las herramientas tienen un fin educativo, por lo que los resultados arrojados por la ejecución de éstas necesitarán una interpretación personal, que dependerá de la carrera que el estudiante curse y la finalidad de la herramienta. Sin embargo, el nivel de conocimiento para el uso de la aplicación es mínimo: bastará un conocimiento básico sobre cómo acceder, iniciar sesión y moverse en una página web.

3.1.4. Entorno operacional

El entorno operacional requerido para la ejecución de la aplicación, con respecto al usuario definido en el apartado anterior, es el siguiente:

- Ordenador con Java instalado y acceso a Internet.
- Navegador con soporte para WebSockets.

3.2. Análisis de requisitos

3.2.1. Requisitos de usuario

En este apartado se especificarán los requisitos de usuario. Estos se dividirán para su planteamiento en:

- Requisitos de capacidad
- Requisitos de restricción

Los requisitos de usuario serán recogidos en tablas con el siguiente formato:

ID	RU-X-YY
Nombre	
Fuente	
Prioridad	
Necesidad	
Verificabilidad	
Estabilidad	
Descripción	

Tabla 1: Requisito de usuario

- **ID:** identificador unívoco del requisito. Su nomenclatura es la siguiente:
 - **RU:** siglas correspondientes a “Requisito de Usuario”.
 - **X:** tipo de requisito. Puede tomar los valores “C” (Capacidad) o “R” (Restricción).
 - **YY:** numeración consecutiva de los requisitos, comenzando por 01.
- **Nombre:** nombre del requisito.
- **Fuente:** origen del requisito. Típicamente, el cliente.
- **Prioridad:** prioridad a la hora de la implementación del proyecto. Puede tomar los valores “Baja”, “Media” o “Alta”.
- **Necesidad:** indicación de cuán necesaria es la inclusión de un requisito en el proyecto. Puede tomar los valores “Esencial”, “Conveniente” u “Opcional”.
- **Verificabilidad:** medida de la facilidad con la que se puede comprobar que el requisito ha sido incorporado al diseño del proyecto. Puede tomar los valores “Baja”, “Media” o “Alta”.
- **Estabilidad:** medida de las modificaciones que puede sufrir el requisito durante el desarrollo del proyecto. Puede tomar los valores “Estable” o “Inestable”.

➤ **Descripción:** descripción detallada del requisito.

3.2.1.1. Requisitos de capacidad

ID	RU-C-01
Nombre	Acceso a la aplicación
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá acceder a la aplicación a través de su nombre de usuario y contraseña.

Tabla 2: Requisito RU-C-01

ID	RU-C-02
Nombre	Lista de reservas
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario, apenas entre en la aplicación, verá la lista actualizada de sus reservas.

Tabla 3: Requisito RU-C-02

ID	RU-C-03
Nombre	Lista de reservas vacía
Fuente	Desarrollador
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Inestable
Descripción	Si el usuario no tiene reservas, la lista mostrará un mensaje informando que no hay reservas para este usuario.

Tabla 4: Requisito RU-C-03

ID	RU-C-04
Nombre	Cuenta regresiva de reservas
Fuente	Desarrollador
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Inestable
Descripción	La lista de reservas mostrará al usuario una cuenta regresiva indicando el tiempo restante para el inicio de las reservas o, en caso de haber comenzado alguna de éstas, el tiempo restante para la finalización (tiempo de reserva).

Tabla 5: Requisito RU-C-04

ID	RU-C-05
Nombre	Estado y fecha de las reservas
Fuente	Cliente
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La lista de reservas mostrará al usuario la fecha reservada y el estado de la reserva: “Lista” o “En espera”.

Tabla 6: Requisito RU-C-05

ID	RU-C-06
Nombre	Eliminación de reservas
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá eliminar reservas de su lista de reservas.

Tabla 7: Requisito RU-C-06

ID	RU-C-07
Nombre	Inicio de reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá acceder a las herramientas reservadas cuando su reserva comience.

Tabla 8: Requisito RU-C-07

ID	RU-C-08
Nombre	Menú
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>El usuario podrá elegir de un menú las siguientes opciones:</p> <ul style="list-style-type: none"> ➤ Mis reservas ➤ Reservar ➤ Cuestionario ➤ Ayuda

Tabla 9: Requisito RU-C-08

ID	RU-C-09
Nombre	Acceso al menú
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El menú de opciones será accesible al usuario en cada una de las páginas de la aplicación.

Tabla 10: Requisito RU-C-09

ID	RU-C-10
Nombre	Cierre de sesión
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá abandonar la aplicación cerrando su sesión.

Tabla 11: Requisito RU-C-10

ID	RU-C-11
Nombre	Ayuda
Fuente	Cliente
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación brindará al usuario contenidos de ayuda para un uso óptimo de sus funcionalidades.

Tabla 12: Requisito RU-C-11

ID	RU-C-12
Nombre	Cuestionario
Fuente	Cliente
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación brindará al usuario la posibilidad de rellenar un cuestionario sobre la práctica realizada.

Tabla 13: Requisito RU-C-12

ID	RU-C-13
Nombre	Repetición del cuestionario
Fuente	Desarrollador
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá repetir el cuestionario tantas veces como desee.

Tabla 14: Requisito RU-C-13

ID	RU-C-14
Nombre	Acceso a calendario
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá acceder a un calendario para la reserva de herramientas a través de la opción: Reservar.

Tabla 15: Requisito RU-C-14

ID	RU-C-15
Nombre	Contenido del calendario
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El calendario de reservas mostrará las franjas de tiempo, en la semana actual, en las que las herramientas a usar en la práctica están ocupadas o libres.

Tabla 16: Requisito RU-C-15

ID	RU-C-16
Nombre	Navegación por el calendario
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá visualizar el contenido del calendario para otras semanas.

Tabla 17: Requisito RU-C-16

ID	RU-C-17
Nombre	Elección de reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá marcar en el calendario el día y la franja de tiempo que desea reservar alguna herramienta.

Tabla 18: Requisito RU-C-17

ID	RU-C-18
Nombre	Confirmación de reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará un mensaje al usuario confirmando que la reserva se hizo o, de lo contrario, un mensaje de error.

Tabla 19: Requisito RU-C-18

ID	RU-C-19
Nombre	Error en la reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará al usuario un mensaje de error si el usuario intenta reservar una fecha y/o franja de tiempo pasada.

Tabla 20: Requisito RU-C-19

ID	RU-C-20
Nombre	Posibilidad de reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá reservar las herramientas para la práctica si éstas están libres en la franja deseada.

Tabla 21: Requisito RU-C-20

ID	RU-C-21
Nombre	Modificación de reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá modificar la fecha/franja de tiempo que ha reservado.

Tabla 22: Requisito RU-C-21

ID	RU-C-22
Nombre	Uso exclusivo de las herramientas reservadas
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Media
Estabilidad	Estable
Descripción	El usuario tendrá acceso exclusivo a las herramientas reservadas durante el intervalo de tiempo especificado en la reserva.

Tabla 23: Requisito RU-C-22

ID	RU-C-23
Nombre	Decisión de acceso a la reserva
Fuente	Desarrollador
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Una vez su reserva haya comenzado y siempre que no haya finalizado, el usuario podrá decidir cuándo acceder a la herramienta.

Tabla 24: Requisito RU-C-23

ID	RU-C-24
Nombre	Configuración del PID
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Al acceder a la reserva, la aplicación permitirá al usuario introducir los valores de los parámetros Proporcional, Integral y Derivativo, para la configuración del PID que enviará datos al motor.

Tabla 25: Requisito RU-C-24

ID	RU-C-25
Nombre	Elección de datos para enviar al PID
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Al acceder a la reserva, la aplicación permitirá al usuario elegir su propio fichero de datos para enviar al PID.

Tabla 26: Requisito RU-C-25

ID	RU-C-26
Nombre	Acceso tras elección de parámetros
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Una vez el usuario introduzca los parámetros a enviar a la herramienta, podrá acceder a ésta.

Tabla 27: Requisito RU-C-26

ID	RU-C-27
Nombre	Video del motor
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Quando el usuario acceda a la herramienta, la aplicación mostrará video en tiempo real del motor y su reacción a los datos enviados.

Tabla 28: Requisito RU-C-27

ID	RU-C-28
Nombre	Gráfica de datos enviados y recibidos por el PID
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Quando el usuario acceda a la herramienta, la aplicación mostrará una gráfica con los datos enviados al PID y la respuesta a cada uno de ellos.

Tabla 29: Requisito RU-C-28

ID	RU-C-29
Nombre	Simultaneidad de video y gráfica
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará al usuario el video y la gráfica simultáneamente.

Tabla 30: Requisito RU-C-29

ID	RU-C-30
Nombre	Envío de datos al PID
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Los datos del fichero de texto elegido por el usuario se enviarán uno a uno.

Tabla 31: Requisito RU-C-30

ID	RU-C-31
Nombre	Actualización de la gráfica
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La gráfica se actualizará con cada envío/respuesta del PID.

Tabla 32: Requisito RU-C-31

ID	RU-C-32
Nombre	Salida de la reserva
Fuente	Cliente
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá detener el funcionamiento de las herramientas (video y gráfica) y salir de la reserva.

Tabla 33: Requisito RU-C-32

ID	RU-C-33
Nombre	Finalización de la reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>Cuando el tiempo de la reserva termine, el usuario podrá decidir si desea guardar los datos y no tendrá más acceso a las herramientas.</p>

Tabla 34: Requisito RU-C-33

ID	RU-C-34
Nombre	Guardado de datos
Fuente	Cliente
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>La aplicación ofrecerá al usuario la posibilidad de guardar los datos recibidos/enviados por el PID en un fichero de texto, en una ruta escogida por el usuario, si:</p> <ul style="list-style-type: none"> ➤ El usuario detiene las herramientas. ➤ La reserva finaliza.

Tabla 35: Requisito RU-C-34

ID	RU-C-35
Nombre	Regreso a la reserva
Fuente	Cliente
Prioridad	Alta
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá volver a utilizar las herramientas si su reserva no ha finalizado.

Tabla 36: Requisito RU-C-35

3.2.1.2. Requisitos de restricción

ID	RU-R-01
Nombre	Soporte para sistemas operativos
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación será accesible desde cualquier sistema operativo.

Tabla 37: Requisito RU-R-01

ID	RU-R-02
Nombre	Internet e instalación de Java
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El ordenador desde el que se acceda a la aplicación deberá tener conexión a Internet y Java instalado.

Tabla 38: Requisito RU-R-02

ID	RU-R-03
Nombre	Soporte para WebSockets
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El navegador que se use para acceder a la aplicación deberá tener soporte para WebSockets.

Tabla 39: Requisito RU-R-03

ID	RU-R-04
Nombre	No registro de usuarios
Fuente	Cliente
Prioridad	Media
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Inestable
Descripción	La aplicación no brindará el servicio de registro de usuarios.

Tabla 40: Requisito RU-R-04

ID	RU-R-05
Nombre	Usuarios en BBDD
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario solo podrá acceder a la aplicación si está registrado en la base de datos de ésta e introduce un usuario y contraseña correctos.

Tabla 41: Requisito RU-R-05

ID	RU-R-06
Nombre	Borrado de reserva comenzada
Fuente	Desarrollador
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Las reservas comenzadas no podrán borrarse de la lista de reservas.

Tabla 42: Requisito RU-R-06

ID	RU-R-07
Nombre	Guardado de cuestionario
Fuente	Cliente
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Baja
Estabilidad	Estable
Descripción	La aplicación solo guardará el último cuestionario que el usuario rellene.

Tabla 43: Requisito RU-R-07

ID	RU-R-08
Nombre	Horarios para las reservas
Fuente	Cliente
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Inestable
Descripción	La aplicación solo permitirá reservar las herramientas de lunes a viernes, en la franja horaria de 9:00 a 21:00.

Tabla 44: Requisito RU-R-08

ID	RU-R-09
Nombre	Reserva de fechas anteriores
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario no podrá reservar fechas anteriores al momento de reservar.

Tabla 45: Requisito RU-R-09

ID	RU-R-10
Nombre	Franja con herramientas ocupadas
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario no podrá reservar en una franja en la que las herramientas estén ocupadas.

Tabla 46: Requisito RU-R-10

ID	RU-R-11
Nombre	Reserva única
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario no podrá tener más de una reserva.

Tabla 47: Requisito RU-R-11

ID	RU-R-12
Nombre	Acceso a reserva restringido
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario solamente podrá acceder a su reserva si ésta ha comenzado.

Tabla 48: Requisito RU-R-12

ID	RU-R-13
Nombre	Acceso a herramientas restringido
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario solamente podrá acceder a las herramientas si las ha reservado antes, la reserva ha comenzado y no finalizado.

Tabla 49: Requisito RU-R-13

ID	RU-R-14
Nombre	Abandono de reserva restringido
Fuente	Cliente
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>El usuario solamente podrá salir de la reserva si:</p> <ul style="list-style-type: none"> ➤ Antes detiene el funcionamiento de las herramientas. ➤ La reserva finaliza.

Tabla 50: Requisito RU-R-14

3.2.2. Casos de uso

En este apartado se definirán los casos de uso de la aplicación. Primeramente se ofrecerá un diagrama y posteriormente se detallará cada caso de uso en tablas.

Los casos de uso son las situaciones en las que puede encontrarse el usuario durante el uso de la aplicación. El siguiente diagrama es una representación gráfica de éstos:

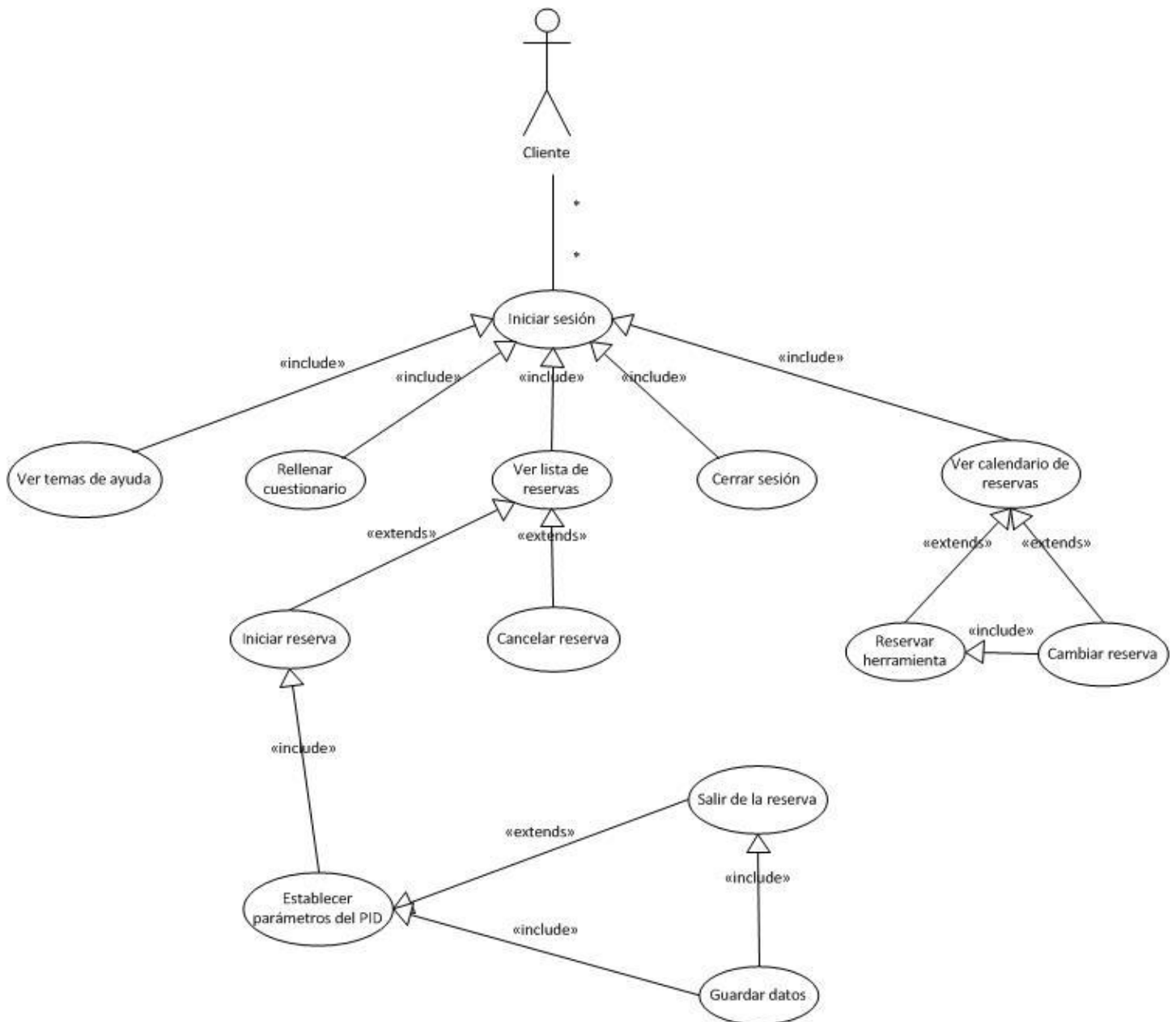


Ilustración 20: Casos de uso

El usuario, en este caso, es la única entidad que interactúa con los casos de uso, por lo que será el único actor. Todas las funcionalidades que ofrece la aplicación serán accesibles solamente tras el inicio de sesión. Si el usuario accede al calendario de reservas, podrá hacer una nueva reserva o cambiar de fecha una reserva previa. Si el usuario accede a la lista de reservas, podrá cancelar (si no han comenzado) o iniciar (si ya han comenzado) las reservas. Para acceder al streaming de video del motor y la

gráfica resultante del envío y recepción de datos del PID, el usuario deberá haber iniciado la reserva previamente y los parámetros de configuración del PID deberán haber sido establecidos. En el caso de que la reserva finalice o el usuario decida salir de ella (la aplicación brindará esta posibilidad), el usuario deberá decidir si desea o no guardar los datos de la gráfica.

Para un análisis más detallado de los casos de uso, éstos serán recogidos en tablas con el siguiente formato:

ID	CU-YY
Nombre	
Actor	
Objetivo	
Precondiciones	
Escenario principal	
Escenario alternativo	
Postcondiciones	

Tabla 51: Caso de uso

- **ID:** identificador unívoco del caso de uso. Su nomenclatura es la siguiente:
 - **CU:** siglas correspondientes a “Caso de Uso”.
 - **YY:** numeración consecutiva de los casos de uso, comenzando por 01.
- **Nombre:** nombre del caso de uso.
- **Actor:** agente que interactúa con el caso de uso.
- **Objetivo:** objetivo que persigue la acción ejercida por el actor.
- **Precondiciones:** condiciones que deben cumplirse antes de la ejecución del caso de uso.

- **Escenario principal:** descripción detallada de los pasos que debe seguir el actor para llevar a cabo el caso de uso.
- **Escenario alternativo:** descripción detallada de los pasos que debe seguir el actor si se origina una bifurcación.
- **Postcondiciones:** estado del sistema tras la correcta ejecución del caso de uso.

ID	CU-01
Nombre	Iniciar sesión.
Actor	Usuario de la aplicación.
Objetivo	Iniciar sesión en la aplicación.
Precondiciones	<ul style="list-style-type: none"> ➤ El ordenador del usuario debe estar conectado a Internet. ➤ Debe haber conexión a la base de datos de la aplicación. ➤ El usuario no debe tener su sesión iniciada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario escribe la dirección de la aplicación en el navegador. 2. El usuario introduce su nombre de usuario y contraseña y pulsa el botón Entrar. 3. La aplicación reconoce al usuario y éste inicia su sesión.
Escenario alternativo	<ol style="list-style-type: none"> 1. El usuario escribe la dirección de la aplicación en el navegador. 2. El usuario introduce su nombre de usuario y contraseña y pulsa el botón Entrar. 3. La aplicación no reconoce al usuario y le muestra un mensaje indicando que el usuario y/o la contraseña son incorrectos.
Postcondiciones	El usuario inicia su sesión en la aplicación.

Tabla 52: Caso de uso CU-01

ID	CU-02
Nombre	Cerrar sesión.
Actor	Usuario de la aplicación.
Objetivo	Cerrar sesión en la aplicación.
Precondiciones	➤ El usuario debe tener su sesión iniciada.
Escenario principal	1. El usuario pulsa Cerrar sesión y sale de la aplicación.
Escenario alternativo	-
Postcondiciones	La sesión del usuario en la aplicación finaliza.

Tabla 53: Caso de uso CU-02

ID	CU-03
Nombre	Ver temas de ayuda.
Actor	Usuario de la aplicación.
Objetivo	Acceder a los temas de ayuda brindados por la aplicación.
Precondiciones	➤ El usuario debe tener su sesión iniciada.
Escenario principal	1. El usuario elige del menú la opción Ayuda y visualiza los temas ofrecidos por la aplicación. 2. El usuario navega por los distintos temas pulsando sobre los títulos.
Escenario alternativo	-
Postcondiciones	El usuario accede a los temas de ayuda ofrecidos por la aplicación.

Tabla 54: Caso de uso CU-03

ID	CU-04
Nombre	Rellenar cuestionario.
Actor	Usuario de la aplicación.
Objetivo	Rellenar y enviar cuestionario sobre la práctica realizada.
Precondiciones	➤ El usuario debe tener su sesión iniciada.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario elige del menú la opción Cuestionario y visualiza las preguntas contenidas en éste. 2. El usuario elige las respuestas para cada pregunta o deja las que están por defecto. 3. El usuario pulsa Enviar, sus respuestas son enviadas y se le muestra un mensaje de confirmación.
Escenario alternativo	<ol style="list-style-type: none"> 1. El usuario elige del menú la opción Cuestionario y visualiza las preguntas contenidas en éste. 2. El usuario elige las respuestas para cada pregunta o deja las que están por defecto. 3. El usuario no envía las respuestas y sale del cuestionario eligiendo alguna otra opción de la aplicación.
Postcondiciones	El usuario rellena el cuestionario y envía sus respuestas.

Tabla 55: Caso de uso CU-04

ID	CU-05
Nombre	Ver calendario de reservas.
Actor	Usuario de la aplicación.
Objetivo	Ver en el calendario la disponibilidad para hacer una reserva.
Precondiciones	➤ El usuario debe tener su sesión iniciada.
Escenario principal	1. El usuario elige del menú la opción Reservar y visualiza el calendario con la disponibilidad para reservar.
Escenario alternativo	-
Postcondiciones	El calendario de reservas se muestra al usuario.

Tabla 56: Caso de uso CU-05

ID	CU-06
Nombre	Reservar herramienta.
Actor	Usuario de la aplicación.
Objetivo	Reservar una herramienta para una fecha determinada.
Precondiciones	<ul style="list-style-type: none"> ➤ El usuario debe tener su sesión iniciada. ➤ El usuario debe haber elegido la opción Reservar y estar visualizando el calendario de reservas.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario hace clic en alguna fecha. 2. La fecha es posterior al momento actual y las herramientas están disponibles para esa fecha. La aplicación reserva la fecha para el usuario y le muestra a éste un mensaje de confirmación.
Escenario alternativo	<ol style="list-style-type: none"> 1. El usuario hace clic en alguna fecha. 2a. La fecha es anterior al momento actual. La aplicación muestra al usuario un mensaje de error específico. 2b. Las herramientas no están disponibles para esa fecha. La aplicación no hace nada y no se produce la reserva.
Postcondiciones	La aplicación reserva la fecha elegida por el usuario.

Tabla 57: Caso de uso CU-06

ID	CU-07
Nombre	Cambiar reserva.
Actor	Usuario de la aplicación.
Objetivo	Cambiar una reserva de una herramienta a otra fecha.
Precondiciones	<ul style="list-style-type: none"> ➤ El usuario debe tener su sesión iniciada. ➤ El usuario debe haber elegido la opción Reservar y estar visualizando el calendario de reservas. ➤ El usuario debe tener alguna reserva. ➤ La reserva elegida no debe haber comenzado.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario hace clic en alguna fecha. 2. La fecha es posterior al momento actual y las herramientas están disponibles para esa fecha. La aplicación elimina la reserva que tenía el usuario, reserva la nueva fecha elegida por éste y muestra un mensaje de confirmación del cambio.
Escenario alternativo	<ol style="list-style-type: none"> 1. El usuario hace clic en alguna fecha. 2a. La fecha es anterior al momento actual. La aplicación muestra al usuario un mensaje de error específico. 2b. Las herramientas no están disponibles para esa fecha. La aplicación muestra al usuario un mensaje de error específico.
Postcondiciones	La aplicación elimina la antigua reserva y reserva la nueva fecha elegida por el usuario.

Tabla 58: Caso de uso CU-07

ID	CU-08
Nombre	Ver lista de reservas.
Actor	Usuario de la aplicación.
Objetivo	Ver la lista de las reservas realizadas y sus estados.
Precondiciones	➤ El usuario debe tener su sesión iniciada.
Escenario principal	1. El usuario elige del menú la opción Mis reservas y visualiza sus reservas y el estado de éstas.
Escenario alternativo	1. El usuario elige del menú la opción Mis reservas. La aplicación le muestra una lista con solo una fila informando que no tiene reservas.
Postcondiciones	El usuario visualiza la lista de sus reservas.

Tabla 59: Caso de uso CU-08

ID	CU-09
Nombre	Cancelar reservas.
Actor	Usuario de la aplicación.
Objetivo	Cancelar una reserva previamente realizada.
Precondiciones	<ul style="list-style-type: none"> ➤ El usuario debe tener su sesión iniciada. ➤ El usuario debe haber elegido la opción Mis reservas y estar visualizando la lista de reservas. ➤ El usuario debe tener alguna reserva. ➤ La reserva elegida no debe haber comenzado.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón Eliminar de la reserva. 2. La aplicación pide al usuario que confirme si quiere eliminar la reserva. 3. El usuario pulsa OK, la reserva desaparece de la lista y la aplicación cancela la reserva.
Escenario alternativo	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón Eliminar de la reserva. 2. La aplicación pide al usuario que confirme si quiere eliminar la reserva. 3. El usuario pulsa Cancelar y no se elimina la reserva.
Postcondiciones	La aplicación cancela la reserva y ésta desaparece de la lista de reservas.

Tabla 60: Caso de uso CU-09

ID	CU-10
Nombre	Iniciar reserva.
Actor	Usuario de la aplicación.
Objetivo	Iniciar una reserva realizada.
Precondiciones	<ul style="list-style-type: none"> ➤ El usuario debe tener su sesión iniciada. ➤ El usuario debe haber elegido la opción Mis reservas y estar visualizando la lista de reservas. ➤ El usuario debe tener alguna reserva. ➤ La reserva elegida debe haber comenzado
Escenario principal	1. El usuario pulsa el botón Comenzar y la aplicación le envía a la página para la configuración de los parámetros del PID.
Escenario alternativo	-
Postcondiciones	El usuario es enviado a la página para la configuración de los parámetros del PID.

Tabla 61: Caso de uso CU-10

ID	CU-11
Nombre	Establecer parámetros de configuración del PID
Actor	Usuario de la aplicación.
Objetivo	Establecer los parámetros de configuración y los datos que se enviarán al PID.
Precondiciones	<ul style="list-style-type: none"> ➤ El usuario debe tener su sesión iniciada. ➤ La reserva del usuario debe haber comenzado. ➤ El usuario debe haber elegido la opción Mis reservas y luego haber pulsado el botón Comenzar.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario introduce los valores (válidos) de los parámetros Proporcional, Integral y Derivativo. 2. El usuario elige un fichero de texto (válido y previamente creado por el propio usuario) con datos para enviar al PID. 3. El usuario pulsa el botón Enviar y accede a la página con el video y la gráfica de datos del motor.
Escenario alternativo	-
Postcondiciones	El usuario visualiza la página con el video del motor y la gráfica resultante del envío y recepción de datos del PID.

Tabla 62: Caso de uso CU-11

ID	CU-12
Nombre	Salir de la reserva.
Actor	Usuario de la aplicación.
Objetivo	Abandonar la reserva.
Precondiciones	<ul style="list-style-type: none"> ➤ El usuario debe tener su sesión iniciada. ➤ La reserva del usuario debe haber comenzado. ➤ El usuario debe haber elegido la opción Mis reservas y luego haber pulsado el botón Comenzar. ➤ El usuario debe haber elegido los parámetros de configuración del PID y estar visualizando el video y la gráfica de datos del PID.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario pulsa Salir. La aplicación detiene el video y el flujo de datos de la gráfica y pregunta al usuario si desea guardar los datos. 2. El usuario pulsa Aceptar o Cancelar y es enviado a la página de Mis reservas.
Escenario alternativo	-
Postcondiciones	Las herramientas se detienen y el usuario es enviado a la página de Mis reservas.

Tabla 63: Caso de uso CU-12

ID	CU-13
Nombre	Guardar datos.
Actor	Usuario de la aplicación.
Objetivo	Guardar los datos procedentes del envío/recepción de datos del PID.
Precondiciones	<ul style="list-style-type: none"> ➤ El usuario debe tener su sesión iniciada. ➤ La reserva del usuario debe haber comenzado. ➤ El usuario debe haber elegido la opción Mis reservas y luego haber pulsado el botón Comenzar. ➤ El usuario debe haber elegido los parámetros de configuración del motor y estar visualizando el video y la gráfica de datos del PID ➤ El tiempo de reserva del usuario debe haber llegado a su fin o el usuario debe haber pulsado el botón Salir. ➤ La aplicación debe haber mostrado un mensaje preguntando si se desean guardar los datos.
Escenario principal	<ol style="list-style-type: none"> 1. El usuario pulsa Aceptar. 2. El usuario elige la ruta para guardar el fichero de datos y pulsa Aceptar. Se guardan los datos y se pasa a la página de Mis reservas.
Escenario alternativo	<ol style="list-style-type: none"> 1. El usuario pulsa Cancelar, no se guardan los datos y es enviado a la página de Mis reservas.
Postcondiciones	Los datos son guardados y el usuario es enviado a la página de Mis reservas.

Tabla 64: Caso de uso CU-13

3.2.3. Requisitos de software

En este apartado se especificarán los requisitos de software que definen, desde el punto de vista del diseño del software, el modo en el que el sistema implementará los requisitos de usuario. Se dividirán para su planteamiento en:

- Requisitos funcionales
- Requisitos no funcionales

Los requisitos de software serán recogidos en tablas con el siguiente formato:

ID	RS-X-YY
Nombre	
Prioridad	
Necesidad	
Verificabilidad	
Estabilidad	
Descripción	
RU implementados	

Tabla 65: Requisito de software

- **ID:** identificador unívoco del requisito. Su nomenclatura es la siguiente:
 - **RS:** siglas correspondientes a “Requisito de Software”.
 - **X:** tipo de requisito. Puede tomar los valores “F” (Funcional) o “N” (No funcional).
 - **YY:** numeración consecutiva de los requisitos, comenzando por 01.
- **Nombre:** nombre del requisito.
- **Prioridad:** prioridad a la hora de la implementación del proyecto. Puede tomar los valores “Baja”, “Media” o “Alta”.

- **Necesidad:** indicación de cuán necesaria es la inclusión de un requisito en el proyecto. Puede tomar los valores “Esencial”, “Conveniente” u “Opcional”.
- **Verificabilidad:** medida de la facilidad con la que se puede comprobar que el requisito ha sido incorporado al diseño del proyecto. Puede tomar los valores “Baja”, “Media” o “Alta”.
- **Estabilidad:** medida de las modificaciones que puede sufrir el requisito durante el desarrollo del proyecto. Puede tomar los valores “Estable” o “Inestable”.
- **Descripción:** descripción detallada del requisito.
- **RU implementados:** indicación de los requisitos de usuario implementados a través del requisito de software.

3.2.3.1 Requisitos funcionales

ID	RS-F-01
Nombre	Conexión a base de datos
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación estará conectada a una base de datos en MySQL a la que accederá para las consultas necesarias.
RU implementados	RU-C-01, RU-C-02, RU-C-06, RU-C-18, RU-C-19, RU-C-20, RU-R-05

Tabla 66: Requisito RS-F-01

ID	RS-F-02
Nombre	Verificación de usuario y contraseña
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación permitirá al usuario acceder con su nombre de usuario y contraseña. Verificará si el usuario está registrado haciendo una consulta a la base de datos.
RU implementados	RU-C-01, RU-R-05

Tabla 67: Requisito RS-F-02

ID	RS-F-03
Nombre	Primera página de la aplicación
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La primera página de la aplicación, tras el inicio de sesión, será la página con las reservas del usuario.
RU implementados	RU-C-02

Tabla 68: Requisito RS-F-03

ID	RS-F-04
Nombre	Consulta de reservas de usuario
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación, tras el inicio de sesión, realizará una consulta a la base de datos y, en base a ésta, mostrará una lista actualizada de las reservas del usuario, incluyendo fecha, estado, y tiempo restante para el inicio o finalización de las reservas (mostrado a través de cuentas regresivas).
RU implementados	RU-C-02, RU-C-03, RU-C-04, RU-C-05

Tabla 69: Requisito RS-F-04

ID	RS-F-05
Nombre	Mensaje en lista de reservas vacía
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Inestable
Descripción	La aplicación mostrará una lista con un mensaje de que no hay reservas, en el caso de que no existan reservas para el usuario.
RU implementados	RU-C-02, RU-C-03

Tabla 70: Requisito RS-F-05

ID	RS-F-06
Nombre	Acciones de las cuentas regresivas
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>Las cuentas regresivas implementadas por la aplicación tendrán asignadas acciones a ejecutar una vez llegadas a 0:</p> <ul style="list-style-type: none"> ➤ Cuenta regresiva de tiempo restante para inicio: mostrar cuenta regresiva de tiempo restante para finalización, poner estado a “Lista” y mostrar botón “Comenzar”. ➤ Cuenta regresiva de tiempo restante para finalización: eliminar reserva.
RU implementados	RU-C-04, RU-C-07, RU-C-35

Tabla 71: Requisito RS-F-06

ID	RS-F-07
Nombre	Botón para eliminar reservas
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará un botón, en cada reserva de la lista, que permitirá eliminar dicha reserva (si no ha comenzado).
RU implementados	RU-C-06

Tabla 72: Requisito RS-F-07

ID	RS-F-08
Nombre	Eliminación total de reservas
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Si una reserva es eliminada, la aplicación la borrará de la base de datos y, consecuentemente, de la lista de reservas del usuario.
RU implementados	RU-C-06

Tabla 73: Requisito RS-F-08

ID	RS-F-09
Nombre	Botón Comenzar
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará un botón “Comenzar” en la fila correspondiente a la reserva, si ésta ha comenzado.
RU implementados	RU-C-07, RU-C-23, RU-C-35

Tabla 74: Requisito RS-F-09

ID	RS-F-10
Nombre	Menú en cada página
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>Cada página de la aplicación (excepto la de ejecución de las herramientas) mostrará un menú en el lateral izquierdo con las siguientes opciones:</p> <ul style="list-style-type: none"> ➤ Mis reservas ➤ Reservar ➤ Cuestionario ➤ Ayuda
RU implementados	RU-C-08, RU-C-9

Tabla 75: Requisito RS-F-10

ID	RS-F-11
Nombre	Ícono de cierre de sesión
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará un ícono para cerrar la sesión en la parte superior derecha.
RU implementados	RU-C-10

Tabla 76: Requisito RS-F-11

ID	RS-F-12
Nombre	Opción Ayuda del menú
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La opción Ayuda del menú de la aplicación mostrará contenidos de ayuda para el usuario.
RU implementados	RU-C-11

Tabla 77: Requisito RS-F-12

ID	RS-F-13
Nombre	Ayuda cargada desde fichero de texto
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Los contenidos de ayuda para el usuario serán cargados por la aplicación desde un fichero de texto.
RU implementados	RU-C-11

Tabla 78: Requisito RS-F-13

ID	RS-F-14
Nombre	Opción Cuestionario del menú
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La opción Cuestionario del menú de la aplicación mostrará una lista de preguntas con varias respuestas entre las cuales el usuario podrá elegir.
RU implementados	RU-C-12

Tabla 79: Requisito RS-F-14

ID	RS-F-15
Nombre	Cuestionario cargado desde fichero de texto
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La lista de preguntas del cuestionario será cargada por la aplicación desde un fichero de texto.
RU implementados	RU-C-12

Tabla 80: Requisito RS-F-15

ID	RS-F-16
Nombre	Respuestas por defecto
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Cada pregunta del cuestionario tendrá una respuesta marcada por defecto.
RU implementados	RU-C-12

Tabla 81: Requisito RS-F-16

ID	RS-F-17
Nombre	Decisión de enviar cuestionario
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá enviar las respuestas marcadas o simplemente salir del cuestionario.
RU implementados	RU-C-12

Tabla 82: Requisito RS-F-17

ID	RS-F-18
Nombre	Guardado de respuestas
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación guardará o sobrescribirá (si existen respuestas guardadas anteriores) las respuestas al cuestionario en la base de datos.
RU implementados	RU-C-12, RU-C-13, RU-R-07

Tabla 83: Requisito RS-F-18

ID	RS-F-19
Nombre	Múltiples envíos de cuestionario
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación permitirá enviar el cuestionario tantas veces como se desee y mostrará un mensaje de confirmación del envío cada vez.
RU implementados	RU-C-13

Tabla 84: Requisito RS-F-19

ID	RS-F-20
Nombre	Opción Reservar del menú
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La opción Reservar del menú de la aplicación mostrará un calendario con la disponibilidad para reservar.
RU implementados	RU-C-14, RU-C-15

Tabla 85: Requisito RS-F-20

ID	RS-F-21
Nombre	Carga de reservas en el calendario
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación hará una consulta a la base de datos para cargar las reservas existentes en el calendario.
RU implementados	RU-C-15

Tabla 86: Requisito RS-F-21

ID	RS-F-22
Nombre	Diferenciación de franjas de reserva
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>El calendario de reservas diferenciará la disponibilidad de reserva del siguiente modo:</p> <ul style="list-style-type: none"> ➤ Franja blanca o amarilla (día actual): las herramientas están libres. Se puede reservar. ➤ Franja roja: las herramientas están reservadas por otro usuario. No se puede reservar. ➤ Franja verde: reserva del propio usuario. No se puede reservar. ➤ Franja gris: reserva iniciada. No se puede reservar.
RU implementados	RU-C-15

Tabla 87: Requisito RS-F-22

ID	RS-F-23
Nombre	Cambio de semana del calendario
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El calendario tendrá, en la parte superior, flechas que permitirán moverse a la vista de la semana anterior o la siguiente.
RU implementados	RU-C-16

Tabla 88: Requisito RS-F-23

ID	RS-F-24
Nombre	Marcado de fecha de reserva
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El usuario podrá marcar con el cursor la fecha elegida en el calendario.
RU implementados	RU-C-17

Tabla 89: Requisito RS-F-24

ID	RS-F-25
Nombre	Consulta de disponibilidad de reserva
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación consultará en la base de datos la disponibilidad de las herramientas para confirmar la reserva.
RU implementados	RU-C-18

Tabla 90: Requisito RS-F-25

ID	RS-F-26
Nombre	Comprobaciones antes de reservar
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>La aplicación comparará si el inicio de la franja elegida es mayor que el momento actual.</p> <ul style="list-style-type: none"> ➤ Si no lo es, se mostrará un error informando que no es posible reservar una fecha pasada. ➤ Si lo es, comprobará si el usuario ya tiene una reserva y ésta ha comenzado. <ul style="list-style-type: none"> • Si ha comenzado, mostrará un mensaje informándolo y no permitirá la reserva. • Si no ha comenzado, cambiará la reserva y mostrará un mensaje informando del cambio.
RU implementados	RU-C-18, RU-C-19, RU-C-20, RU-R-09, RU-R-10

Tabla 91: Requisito RS-F-26

ID	RS-F-27
Nombre	Sobrescritura de reserva
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Si el usuario marca una franja, la aplicación consultará en la BBDD si este usuario tiene ya una reserva. Si la franja es distinta a la de la antigua reserva y se confirma la nueva, se borrará la antigua y se guardará la nueva en la BBDD. El calendario se actualizará.
RU implementados	RU-C-21, RU-R-11

Tabla 92: Requisito RS-F-27

ID	RS-F-28
Nombre	Control de exclusividad de reserva
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación controlará durante la reserva de herramientas que, si se asigna una franja a un usuario, el resto de usuarios solo pueda reservar una franja distinta.
RU implementados	RU-C-22

Tabla 93: Requisito RS-F-28

ID	RS-F-29
Nombre	Página de elección de parámetros
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>Cuando el usuario inicie la reserva con el botón Comenzar, la aplicación mostrará una página con cuatro inputs que el usuario deberá rellenar con datos válidos:</p> <ul style="list-style-type: none"> ➤ Proporcional ➤ Integral ➤ Derivativo ➤ Fichero de datos (input de tipo file)
RU implementados	RU-C-24, RU-C-25

Tabla 94: Requisito RS-F-29

ID	RS-F-30
Nombre	Proporcional, Integral y Derivativo válidos
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Los inputs Proporcional, Integral y Derivativo deberán ser rellenados con valores numéricos. No se permitirá dejarlos vacíos.
RU implementados	RU-C-24

Tabla 95: Requisito RS-F-30

ID	RS-F-31
Nombre	Fichero de texto válido
Prioridad	Media
Necesidad	Conveniente
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El fichero de texto elegido y elaborado por el usuario deberá contener 100 números, uno por cada línea.
RU implementados	RU-C-25

Tabla 96: Requisito RS-F-31

ID	RS-F-32
Nombre	Enviar parámetros
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará un botón Enviar para enviar el fichero de datos y los parámetros de configuración del PID.
RU implementados	RU-C-26

Tabla 97: Requisito RS-F-32

ID	RS-F-33
Nombre	Página de ejecución de herramientas
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>Cuando el usuario envíe los datos de configuración del PID, la aplicación mostrará una página con varias zonas:</p> <ul style="list-style-type: none"> ➤ Zona de video: mostrará video en tiempo real del motor y su reacción a los datos enviados. ➤ Zona de gráfica: mostrará una gráfica con los datos enviados al PID y la respuesta a cada uno de ellos. ➤ Zona de cuenta regresiva: mostrará el tiempo de reserva restante. ➤ Zona de datos: mostrará el dato enviado en el momento y la respuesta correspondiente. Se actualizará con cada envío. ➤ Zona de salida de reserva: mostrará un botón Salir, que permitirá abandonar la reserva.
RU implementados	RU-C-27, RU-C-28, RU-C-29

Tabla 98: Requisito RS-F-33

ID	RS-F-34
Nombre	Streaming de video del motor
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación ejecutará la herramienta RLF_Cam para activar con VLC Player la cámara desde la que se observa el motor. Usará el protocolo RTP para el streaming de video en la página.
RU implementados	RU-C-27

Tabla 99: Requisito RS-F-34

ID	RS-F-35
Nombre	Gráfica de datos
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación ejecutará la herramienta RLF_Chart para comenzar el envío y recibo de datos del PID. Se enviarán los datos del fichero de texto elegido y se mostrarán las respuestas del PID en la gráfica.
RU implementados	RU-C-28

Tabla 100: Requisito RS-F-35

ID	RS-F-36
Nombre	Datos enviados uno a uno
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación enviará un dato del fichero de texto al PID, recibirá la respuesta de éste, y continuará con el siguiente dato.
RU implementados	RU-C-30

Tabla 101: Requisito RS-F-36

ID	RS-F-37
Nombre	Dibujo de gráfica
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación irá dibujando en la gráfica dos líneas, añadiendo uno a uno los datos enviados en una línea y las respuestas en la otra.
RU implementados	RU-C-31

Tabla 102: Requisito RS-F-37

ID	RS-F-38
Nombre	Parada de herramientas
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará un botón Salir, que detendrá el funcionamiento de las herramientas y preguntará al usuario si desea guardar los datos de la gráfica.
RU implementados	RU-C-32

Tabla 103: Requisito RS-F-38

ID	RS-F-39
Nombre	Inhabilitación de opciones
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación inhabilitará las otras opciones de la página, de modo que el usuario solamente pueda salir de la reserva utilizando el botón Salir o esperando a la finalización del tiempo de reserva.
RU implementados	RU-C-32, RU-R-14

Tabla 104: Requisito RS-F-39

ID	RS-F-40
Nombre	Mensaje de guardado de datos
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Si el usuario elige salir de la aplicación o la reserva termina por sí misma (la cuenta regresiva llega a 0), se mostrará un mensaje al usuario preguntando si desea guardar los datos de la gráfica.
RU implementados	RU-C-32, RU-C-33, RU-C-34

Tabla 105: Requisito RS-F-40

ID	RS-F-41
Nombre	Fin de tiempo de reserva
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	Cuando termine el tiempo de reserva, la aplicación detendrá el funcionamiento de las herramientas, borrará la reserva de la base de datos y preguntará al usuario si desea guardar los datos de la gráfica.
RU implementados	RU-C-33, RU-C-34

Tabla 106: Requisito RS-F-41

ID	RS-F-42
Nombre	Decisión de guardado de datos
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	<p>La aplicación mostrará un mensaje preguntando si se desea guardar los datos.</p> <ul style="list-style-type: none"> ➤ Aceptar: se creará un fichero cuya ruta puede ser especificada por el usuario. El nombre del fichero será asignado por la propia aplicación. Será un .txt de nombre Usuario+Fecha. ➤ Cancelar: no se guardarán los datos. <p>En cualquier caso, se enviará al usuario a la página de Mis reservas.</p>
RU implementados	RU-C-34

Tabla 107: Requisito RS-F-42

3.2.3.2 Requisitos no funcionales

ID	RS-N-01
Nombre	Aplicación multiplataforma
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación será multiplataforma.
RU implementados	RU-R-01

Tabla 108: Requisito RS-N-01

ID	RS-N-02
Nombre	Aplicación web
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación será de tipo web y estará desarrollada en Java.
RU implementados	RU-R-02

Tabla 109: Requisito RS-N-02

ID	RS-N-03
Nombre	Uso de Websockets
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La comunicación entre el cliente y el servidor se efectuará a través de WebSockets. Se mostrará un mensaje de error si el navegador usado por el usuario no es compatible con WebSockets.
RU implementados	RU-R-03

Tabla 110: Requisito RS-N-03

ID	RS-N-04
Nombre	Inexistencia de página de registro
Prioridad	Media
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación no proporcionará ninguna página para el registro de usuarios.
RU implementados	RU-R-04

Tabla 111: Requisito RS-N-04

ID	RS-N-05
Nombre	Error en inicio de sesión
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación mostrará un mensaje de error si el usuario y contraseña introducidos para el inicio de sesión no se encuentran en la base de datos.
RU implementados	RU-R-05

Tabla 112: Requisito RS-N-05

ID	RS-N-06
Nombre	Borrado del botón Eliminar
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación no mostrará/suprimirá el botón Eliminar para las reservas comenzadas.
RU implementados	RU-R-06

Tabla 113: Requisito RS-N-06

ID	RS-N-07
Nombre	Franjas de reserva
Prioridad	Baja
Necesidad	Opcional
Verificabilidad	Alta
Estabilidad	Estable
Descripción	El calendario de reservas no mostrará los fines de semana y mostrará, para cada día, las horas de 9:00 a 21:00 divididas en intervalos de 15 minutos.
RU implementados	RU-R-08

Tabla 114: Requisito RS-N-07

ID	RS-N-08
Nombre	Dependencia de cuenta regresiva
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La lista de reservas no ofrecerá ningún modo de acceso a la reserva hasta que la cuenta regresiva del tiempo restante para el inicio de reserva llegue a 0.
RU implementados	RU-R-12, RU-R-13

Tabla 115: Requisito RS-N-08

ID	RS-N-09
Nombre	Borrado de reserva tras finalización
Prioridad	Alta
Necesidad	Esencial
Verificabilidad	Alta
Estabilidad	Estable
Descripción	La aplicación borrará la reserva de la base de datos una vez finalizada.
RU implementados	RU-R-13

Tabla 116: Requisito RS-N-09

3.2.4. Matriz de trazabilidad

Tomando como base el campo “RU implementados” de las tablas de requisitos de software, se comprobará que cada requisito de usuario esté implementado por al menos un requisito de software. La siguiente matriz de trazabilidad expresa la relación entre RU y RS:

[illegible]

Tabla 117: Matriz de trazabilidad RU-R

Del análisis de la matriz de trazabilidad se puede concluir, dado que cada RU tiene asociado al menos un RS, que la aplicación implementa todos los requisitos de usuario

4. Diseño

En este capítulo se detallarán los aspectos relacionados con el diseño del sistema. Se especificará el diseño de la arquitectura del sistema, la base de datos y las interfaces de usuario.

4.1. Arquitectura del sistema

El sistema estará desarrollado sobre una arquitectura cliente-servidor. Esta arquitectura es un modelo de aplicación distribuida en el cual la máquina que hace de servidor administra los recursos a ser utilizados por el cliente. Los clientes están conectados a un servidor a través de una red por la cual envían peticiones a éste. El servidor procesa las peticiones y envía respuestas a cada petición de cada cliente.

Para este proyecto el funcionamiento será el siguiente:

- La aplicación cliente envía peticiones al servidor.
- El servidor, en dependencia del tipo de petición, hace consultas a la base de datos y/o se conecta a laboratorios remotos.
 - I. El servidor recibe las respuestas a las consultas y las devuelve a la aplicación cliente (si fuese necesario).
 - II. Se establece la conexión entre servidor y laboratorio y comienza el uso de las herramientas, cuyos flujos de datos (si los hubiese) se envían al cliente a través del servidor.

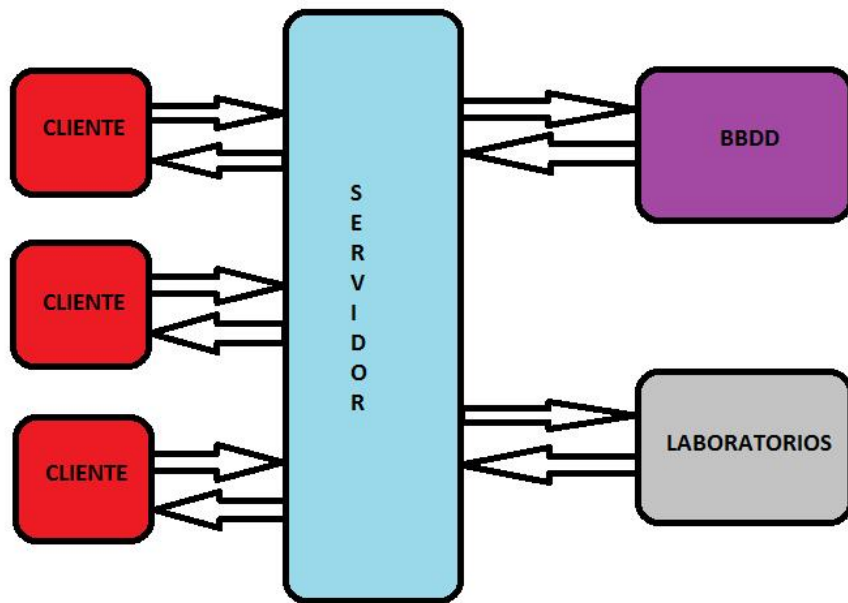


Ilustración 21: Flujo de la aplicación

El cliente podrá realizar peticiones al servidor de distinta índole: inserciones, actualizaciones, recuperación y borrado de datos en la BBDD, y/o peticiones de acceso a herramientas de laboratorios remotos.

El siguiente diagrama representa la relación entre los componentes de la aplicación:

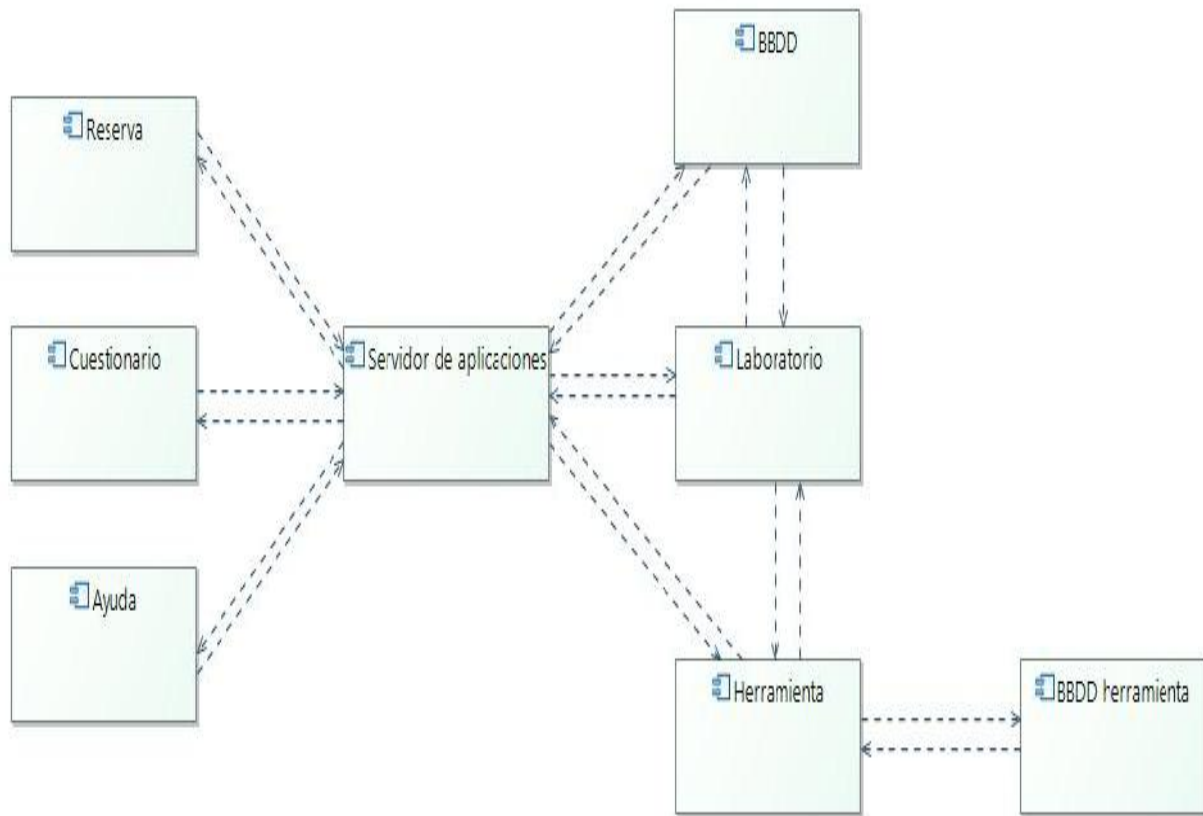


Ilustración 22: Diagrama de componentes [27]

Los servicios que ofrece la aplicación serán gestionados por el servidor de aplicaciones que, en dependencia de la petición que se le envíe, podrá acceder a la base de datos central y/o conectarse al laboratorio. El servidor usará la misma base de datos que el laboratorio para conectarse a las herramientas que vayan a ser usadas por el usuario. Por su parte, dichas herramientas utilizarán su propia base de datos (SQLite) y, una vez que el laboratorio haya establecido la conexión entre las herramientas y el servidor de aplicaciones, éste último podrá comunicarse directamente con las herramientas.

4.2. Base de datos

De las peticiones que recibirá el servidor, muchas de ellas serán consultas, inserciones/actualizaciones y borrados de datos relativos al usuario y sus reservas. Todos estos datos serán gestionados a través de una base de datos MySQL. Esta base de datos será común al servidor y los laboratorios dado que, si se añade/elimina una herramienta o se modifica algún parámetro en los laboratorios, el servidor debe tener esta información actualizada.

En el proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha, la base de datos de los laboratorios respondía a la siguiente estructura:

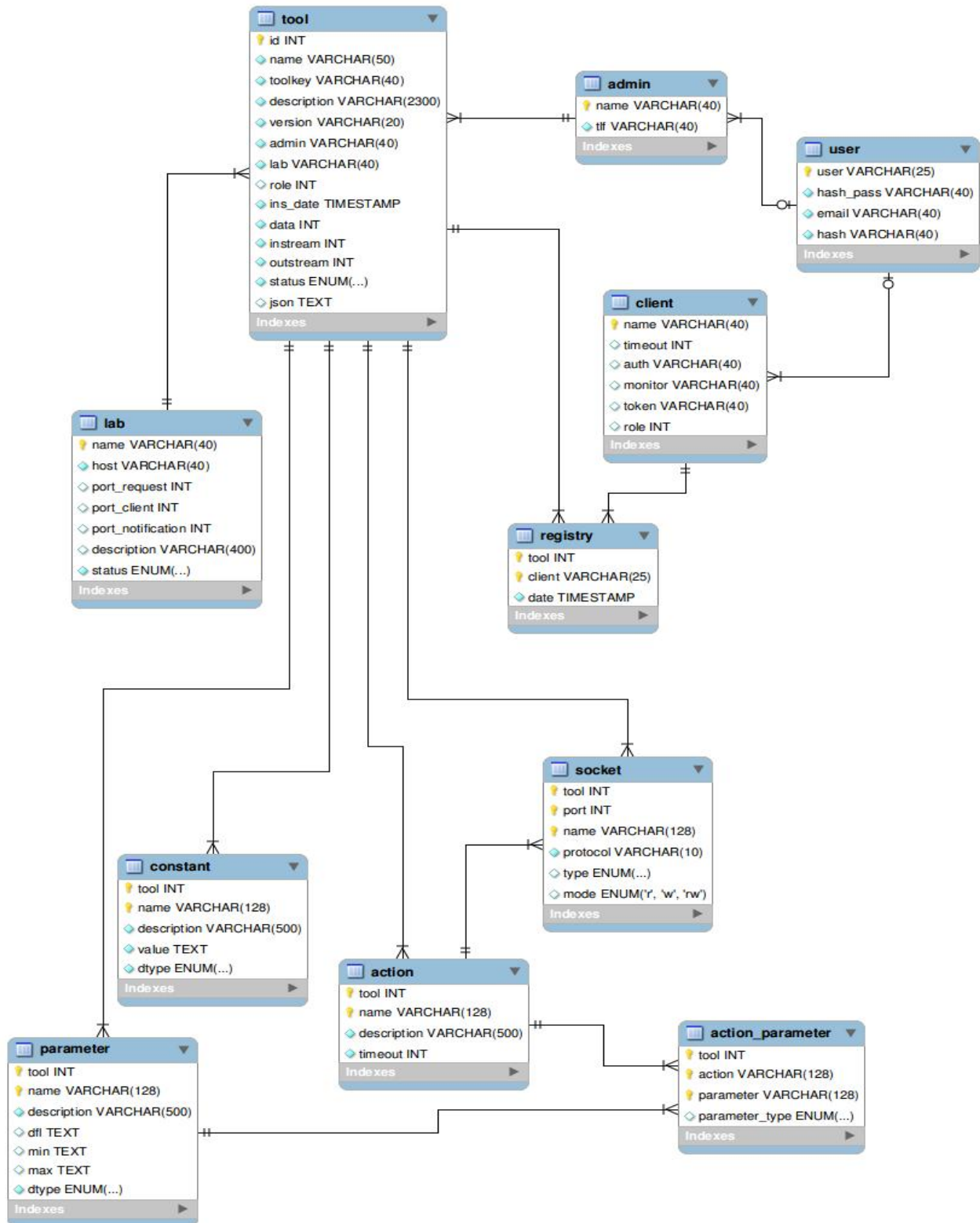


Ilustración 23: Base de datos del laboratorio [1]

En este proyecto se han añadido las tablas *booking*, *test* y *chart_pair*.

- **booking:** almacena las reservas de usuario. Su clave primaria será *tool+user*.
 - *tool*: herramienta reservada (referencia a la tabla *tool*).
 - *user*: usuario que ha reservado la herramienta (referencia a la tabla *user*).
 - *date*: fecha para la que el usuario ha reservado la herramienta.
- **test:** almacena las preguntas y respuestas del cuestionario enviado por el usuario. Su clave primaria será *user+question*.
 - *user*: usuario que ha respondido la pregunta (referencia a la tabla *user*).
 - *question*: pregunta respondida por el usuario.
 - *answer*: respuesta dada por el usuario a la pregunta.
- **chart_pair:** almacena los pares de valores correspondientes al envío y recepción de datos del PID. Estos datos son los que se van dibujando en la gráfica. Cada vez que se envía y recibe un dato, este par se guarda por si el usuario quisiese pasarlos a un fichero de datos luego. Los pares se borran cada vez que el usuario ejecuta la reserva, para comenzar a guardar nuevos pares de esa ejecución. La clave primaria será un identificador unívoco, asignado por la base de datos en la medida que las entradas de la tabla aumentan.
 - *id*: identificador unívoco del par.
 - *user*: usuario que ha enviado y recibido el par (referencia a la tabla *user*).
 - *sent_value*: valor enviado al PID
 - *received_value*: valor recibido del PID.

El siguiente diagrama muestra las relaciones entre las nuevas entidades y las que existían previamente:

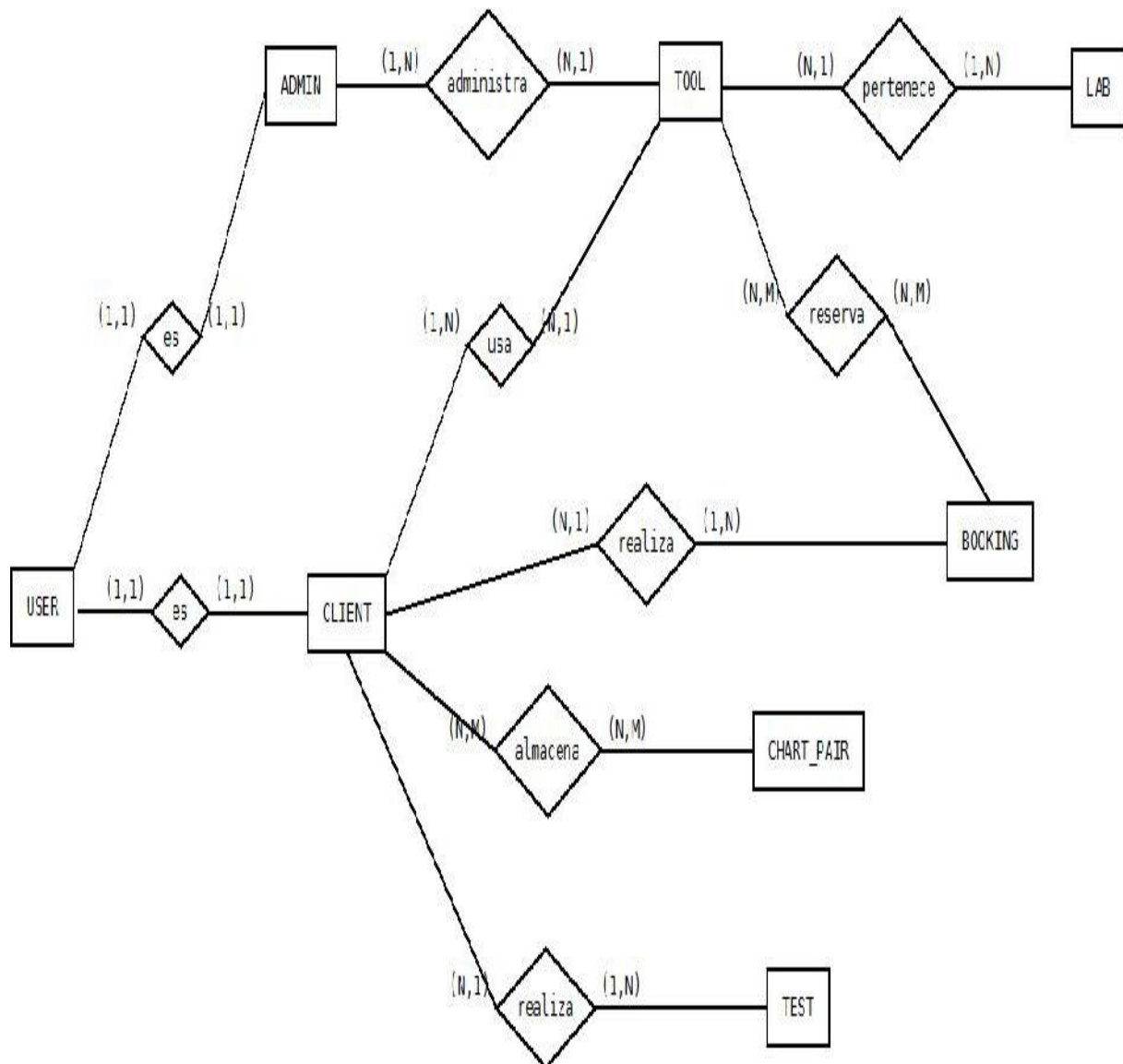


Ilustración 24: Diagrama Entidad-Relación

4.3. Interfaces de usuario

Las interfaces de usuario han sido diseñadas de modo que resulten intuitivas y de fácil uso al usuario. Así mismo, se ha intentado no recargarlas demasiado para evitar molestias visuales. Siendo la aplicación para un uso muy específico y no comercial, el número de interfaces es pequeño y no se ha hecho hincapié en un efecto visual excesivamente impactante. A continuación se detallarán las interfaces con las que cuenta la aplicación.

4.3.1. Inicio de sesión

La página de inicio de sesión será la primera página a la que podrá acceder el usuario. El acceso por el usuario al resto de servicios de la aplicación dependerá de un inicio de sesión exitoso. Esta página controlará que el usuario que acceda a la aplicación esté registrado previamente en la base de datos. Como la aplicación será usada por estudiantes, no se ofrecerá la posibilidad de registrarse, sino que los usuarios registrados previamente en la BBDD serán los estudiantes del centro en que se use la aplicación.

La página mostrará un panel de acceso con dos campos de texto y un botón. En los dos campos de texto se podrá introducir los datos correspondientes a nombre de usuario y contraseña. Una vez pulsado el botón *Entrar*, el usuario:

- Será reconocido por la aplicación y se le enviará a la página con sus reservas (interfaz *Listado de reservas*), en caso de estar registrado en la BBDD.
- Visualizará un mensaje de error (pop-up) indicando que su usuario y/o contraseña no son correctos, en caso de no estar registrado en la BBDD.

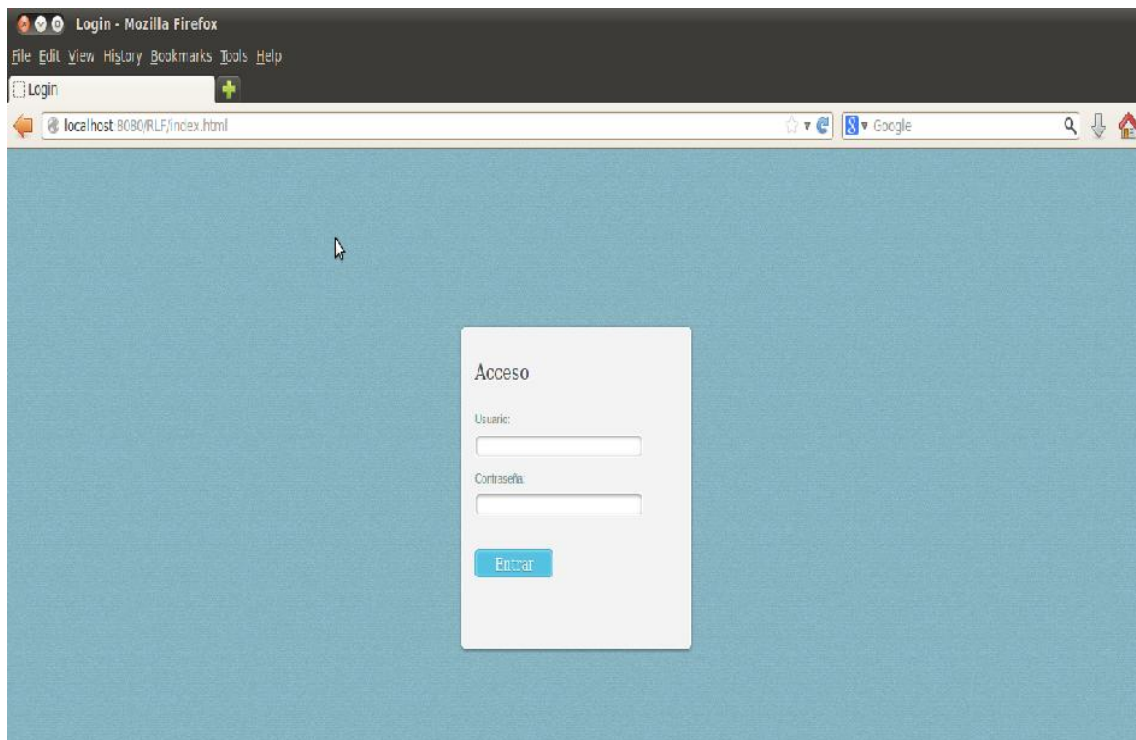


Ilustración 25: Inicio de sesión

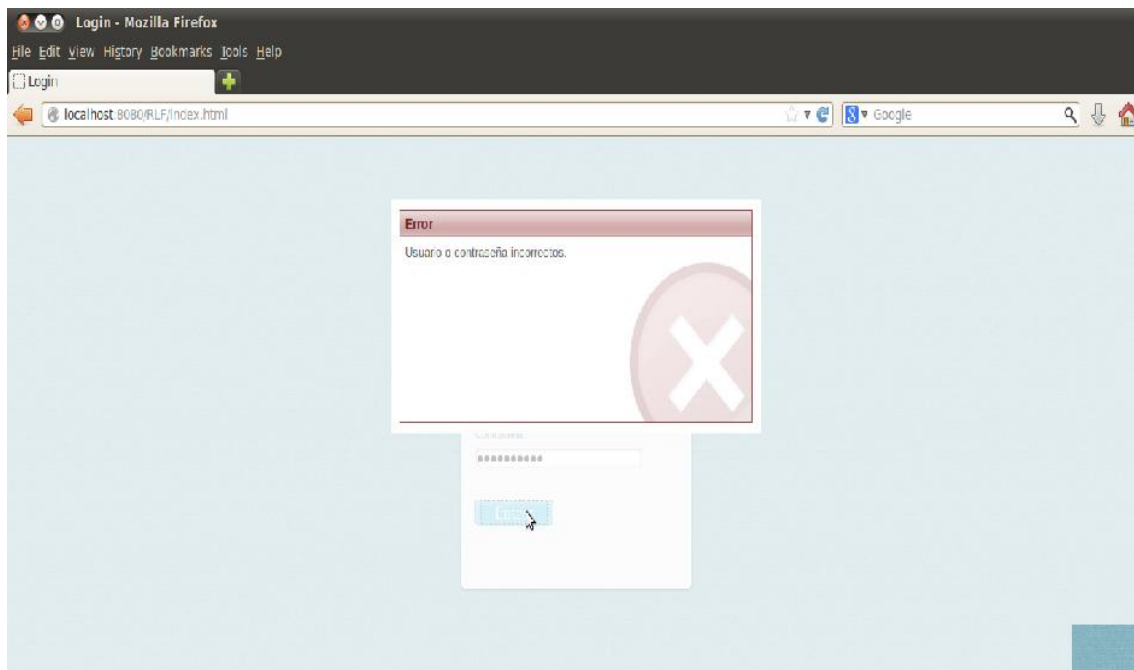


Ilustración 26: Error en el inicio de sesión

4.3.2. Listado de reservas

Si el usuario inicia sesión correctamente, esta será la página de inicio de su sesión. En ella se mostrará el listado de reservas hechas por el usuario. La página estará dividida en dos secciones: un menú y la lista de reservas.

El menú estará ubicado a la izquierda y mostrará las opciones *Reservar*, *Mis reservas*, *Cuestionario* y *Ayuda*. Estas opciones, al ser pulsadas, enviarán al usuario a las interfaces *Listado de reservas*, *Calendario de reservas*, *Cuestionario* y *Ayuda*, respectivamente. De este modo, el usuario podrá navegar entre páginas. Este menú estará presente en todas la interfaces, excepto *Ejecución de la reserva*.

La lista de reservas se ubicará a la derecha y mostrará una tabla con las siguientes columnas:

- **Fecha:** mostrará la fecha (día, mes y año) en que comenzará la reserva en el formato dd/mm/aaaa.
- **Hora:** mostrará la hora en que comenzará la reserva en el formato hh:mm

- **Tiempo restante para inicio:** mostrará una cuenta regresiva que indicará el tiempo restante para el inicio de la reserva. Cuando llegue a 0, la columna *Comenzar* eliminará el botón *Eliminar* (no se podrá cancelar la reserva si ésta ha comenzado) y lo remplazará por un botón *Comenzar*, que permitirá acceder a la reserva. Así mismo, cuando llegue a 0 la cuenta regresiva, se iniciará la cuenta regresiva de la columna *Tiempo restante de uso* y la columna *Estado* mostrará el valor *Lista*.
- **Tiempo restante de uso:** mostrará inicialmente (si la reserva no ha comenzado) el valor 15. Una vez la cuenta regresiva de la columna *Tiempo restante para inicio* llegue a 0, este valor será remplazado por una cuenta regresiva que indicará el tiempo restante de reserva (tiempo que falta para la finalización). Cuando llegue a 0, la reserva será eliminada de la lista y no se podrá acceder más a ella.
- **Estado:** mostrará el valor *En espera* si la reserva aún no ha comenzado o el valor *Lista* en caso contrario.
- **Comenzar:** mostrará un botón *Eliminar* si la reserva aún no ha comenzado o un botón *Comenzar* en caso contrario. El botón *Eliminar* permitirá cancelar la reserva. Pedirá confirmación al usuario antes de realizar esta acción, tras lo cual, en caso afirmativo, la reserva será borrada de la BBDD y del listado inmediatamente. En caso negativo, no se realizará ninguna acción. El botón *Comenzar* enviará al usuario a la interfaz *Configuración del motor*.

Si el usuario no tiene reservas, la tabla contendrá un texto indicando al usuario que no las tiene.

Por otro lado, en la parte superior derecha se indicará el nombre de usuario cuya sesión se ha iniciado y se mostrará una imagen de OFF, cuyo pulsado permitirá al usuario cerrar su sesión. El nombre del usuario y la posibilidad de cerrar sesión se mostrarán en cada una de las páginas de la aplicación.

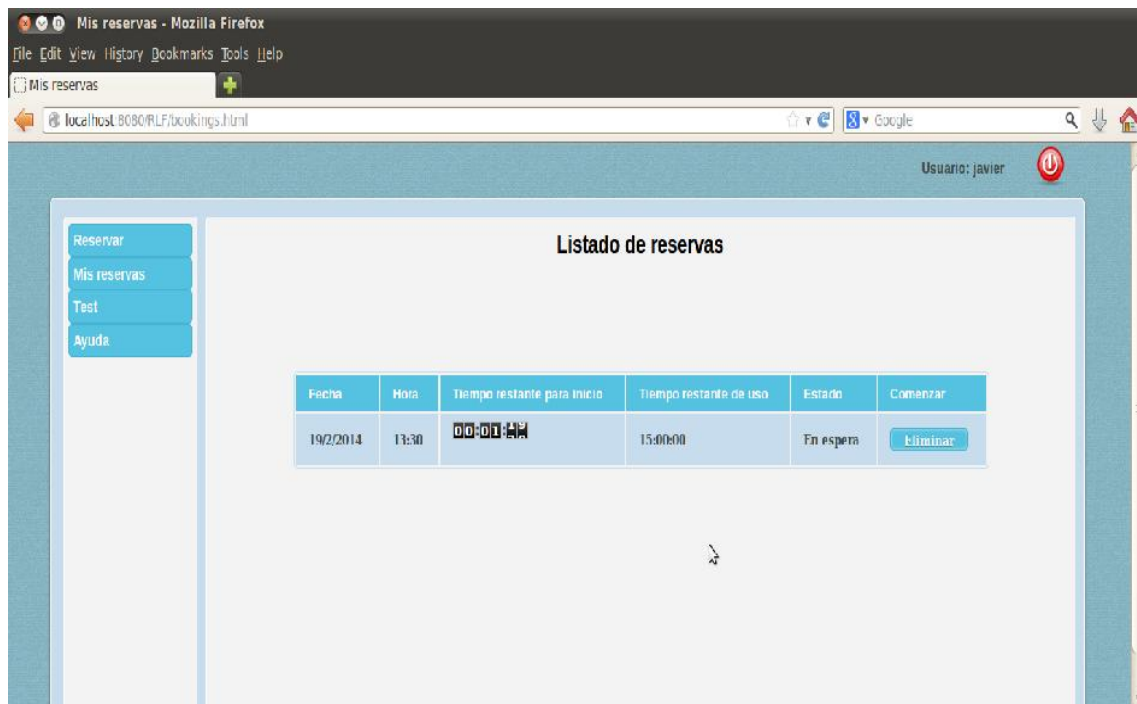


Ilustración 27: Reserva en espera

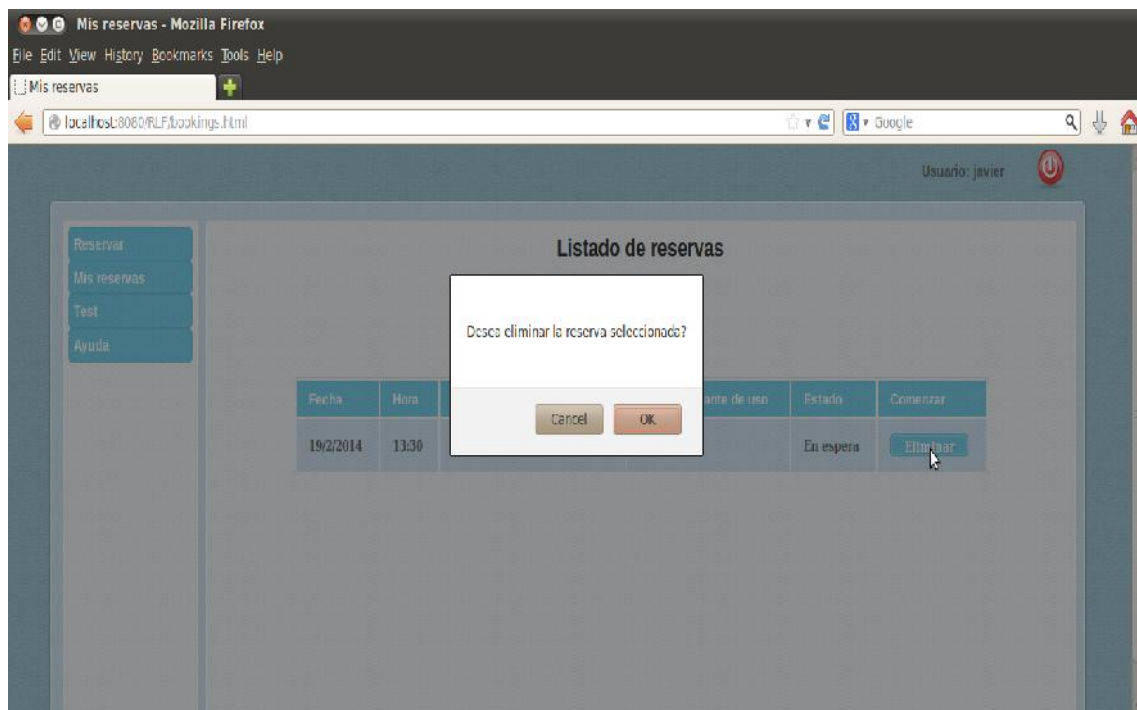


Ilustración 28: Eliminación de reserva

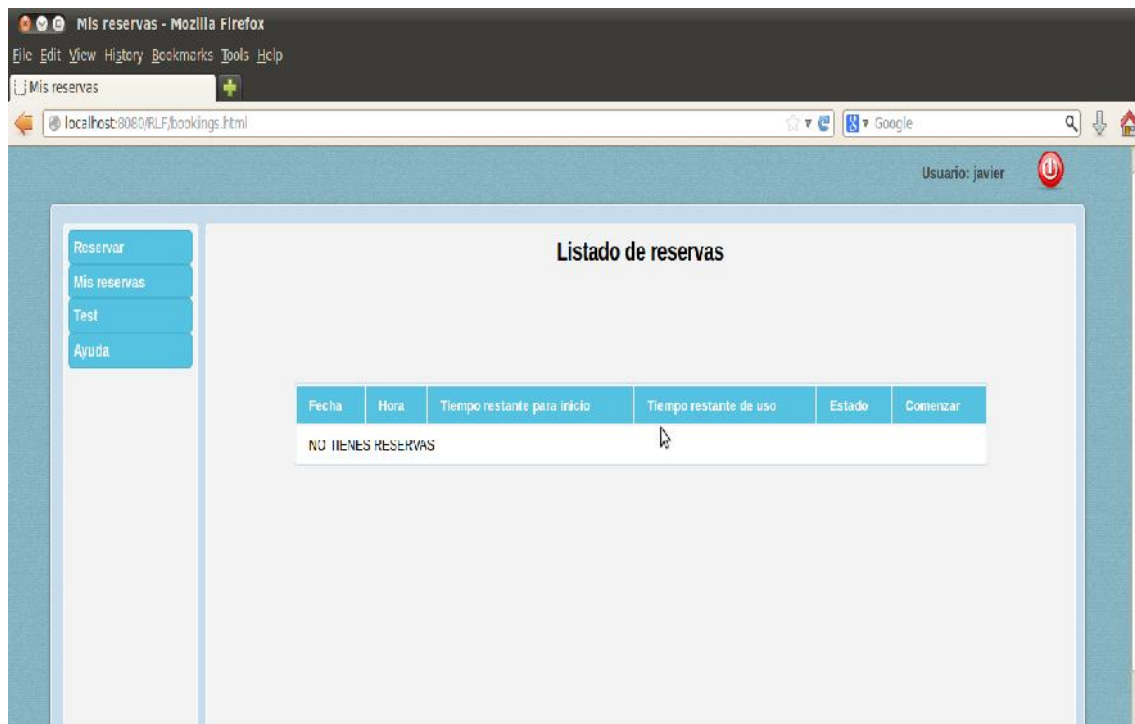


Ilustración 29: Lista vacía tras la eliminación de reserva

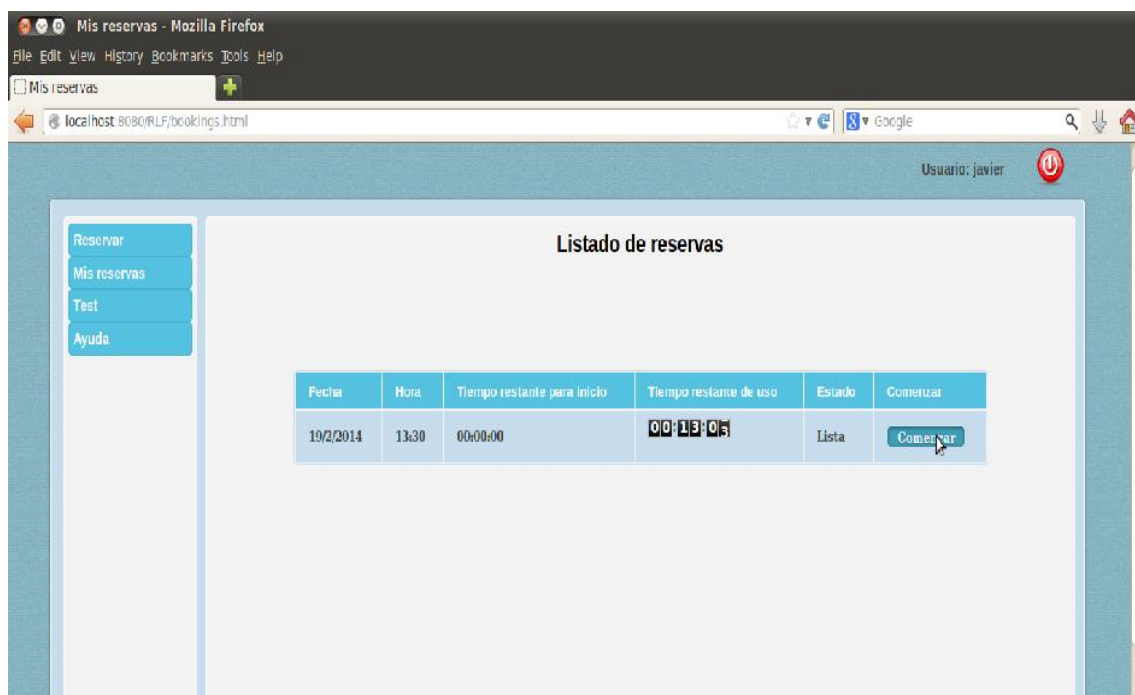


Ilustración 30: Reserva lista

4.3.3. Calendario de reservas

El usuario accederá a esta interfaz a través de la opción *Reservar* del menú que se encuentra en cada página. Mostrará un calendario que indicará la disponibilidad de reserva. En la parte superior derecha del calendario se podrá escoger entre dos vistas: *Semana* y *Día*.

La vista *Semana* (vista por defecto) mostrará un calendario de lunes a viernes de 9 a 21 horas, correspondiente a los días laborables de una universidad. Las horas del día estarán divididas en rangos de 15 minutos, correspondientes al tiempo de reserva. Los días de la semana se mostrarán en la línea superior del calendario y seguirán el formato *Día de la semana Mes/Día* (Ejemplo: Lunes 2/3). Las horas del día se mostrarán en la línea izquierda del calendario y seguirán el formato de 12 horas, am o pm (Ejemplo: 9am). La columna correspondiente al día actual se diferenciará del resto por ser de color amarillo.

El usuario podrá elegir alguna de las franjas de 15 minutos, dándose los siguientes casos:

- **La franja elegida es de color blanco:** no hay reservas en esa fecha. Pueden darse dos subcasos:
 - **La franja es de una fecha pasada*:** se mostrará un mensaje al usuario indicando que no puede reservar una fecha pasada.
 - **La franja es de una fecha futura:** si el usuario no tiene ninguna reserva previa, se mostrará un mensaje al usuario confirmando que la reserva se ha realizado correctamente. Si el usuario ya tiene una reserva, se mostrará un mensaje al usuario indicando el cambio de fecha de la reserva.

En cualquier caso, si el usuario tiene una reserva y ésta ha comenzado, se le indicará con un mensaje que debe esperar a la finalización de su reserva. Esto evita la multiplicidad de reserva de una misma herramienta.

*Si una fecha es pasada o no, está determinado por la fecha y hora de inicio de la franja.
Ejemplo: son las 10:46 y el usuario elige la franja de 10:45-11:00 de ese mismo día. Se considera que el usuario no puede reservar esa franja.

- **La franja elegida es de color verde:** la franja ya está reservada por el usuario. No ocurre nada.
- **La franja elegida es de color rojo:** la franja está reservada por otro usuario. No ocurre nada.
- **La franja elegida es de color gris:** la franja está reservada y la reserva ha comenzado. No ocurre nada.

Todos los mensajes se mostrarán a través de pop-ups de color verde en caso de confirmación y rojo en caso de error.

La vista *Día* tendrá las mismas características y funcionalidad que la vista *Semana*, con excepción de que solamente mostrará el calendario para el día actual.

En la parte superior izquierda del calendario se mostrarán dos flechas y la opción *Hoy*. Las flechas se podrán utilizar para la navegación por el calendario, avanzando a la semana/día (dependiendo de la vista elegida) siguiente o anterior. La opción *Hoy* enviará al usuario de vuelta a la vista de la semana/día actual.

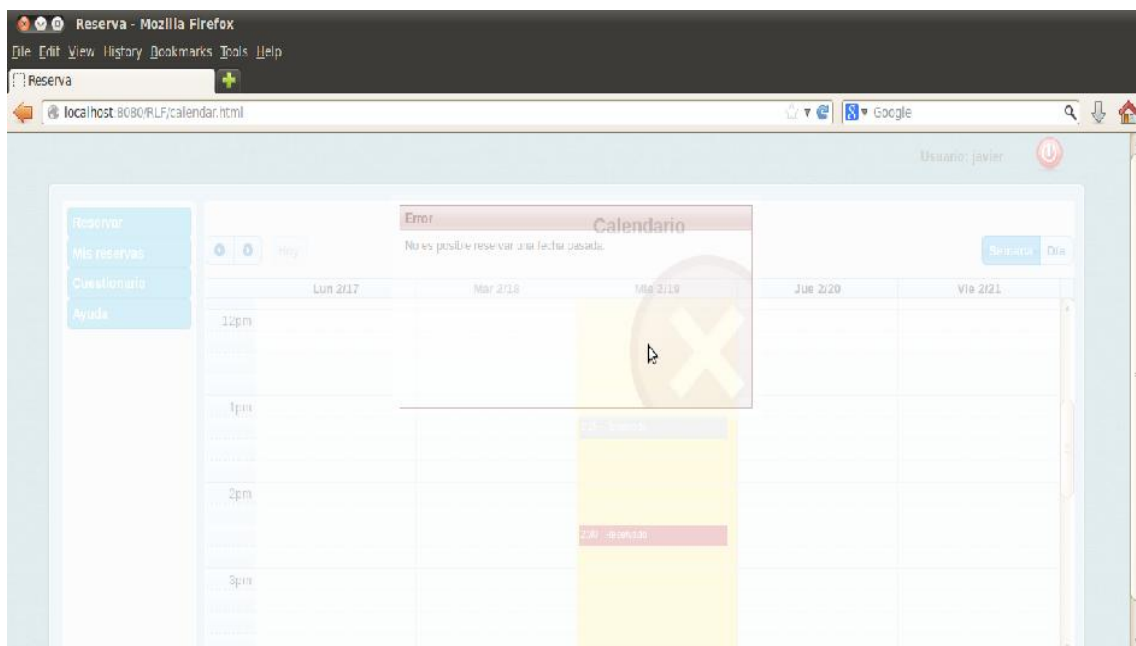


Ilustración 31: Intento de reserva de fecha pasada

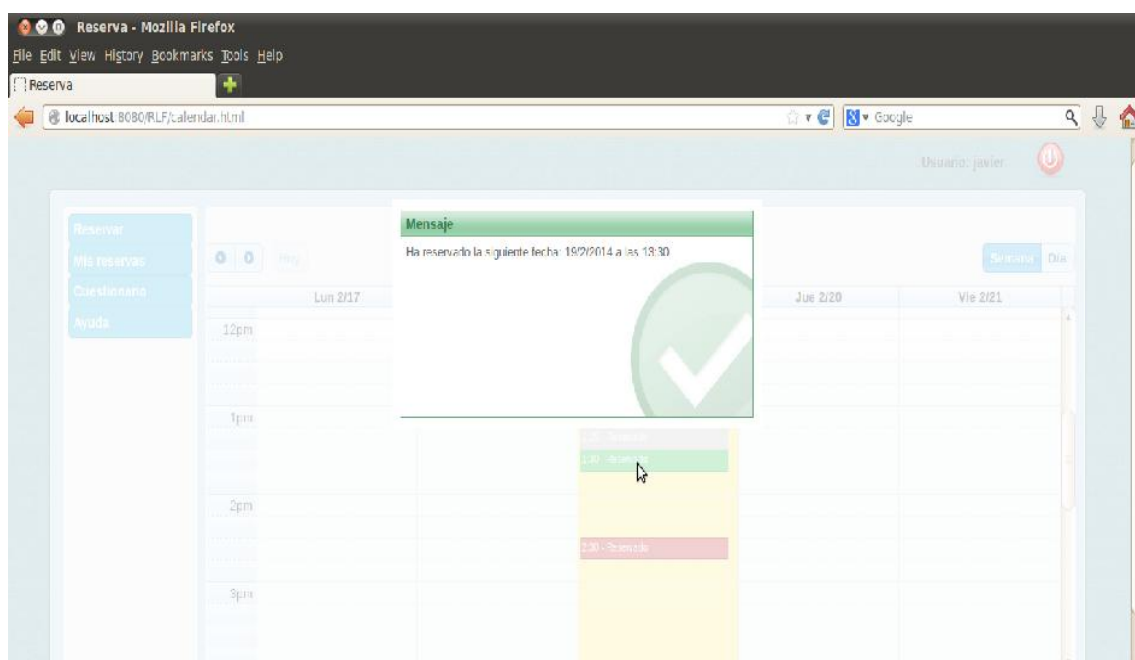


Ilustración 32: Confirmación de reserva

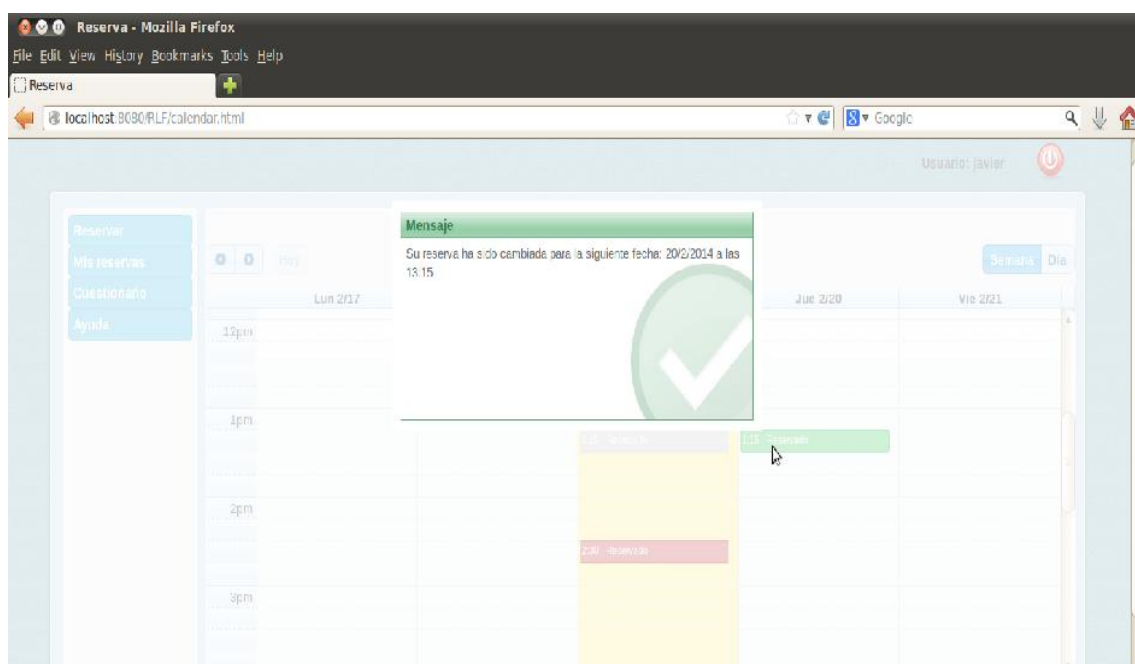


Ilustración 33: Confirmación de cambio de reserva

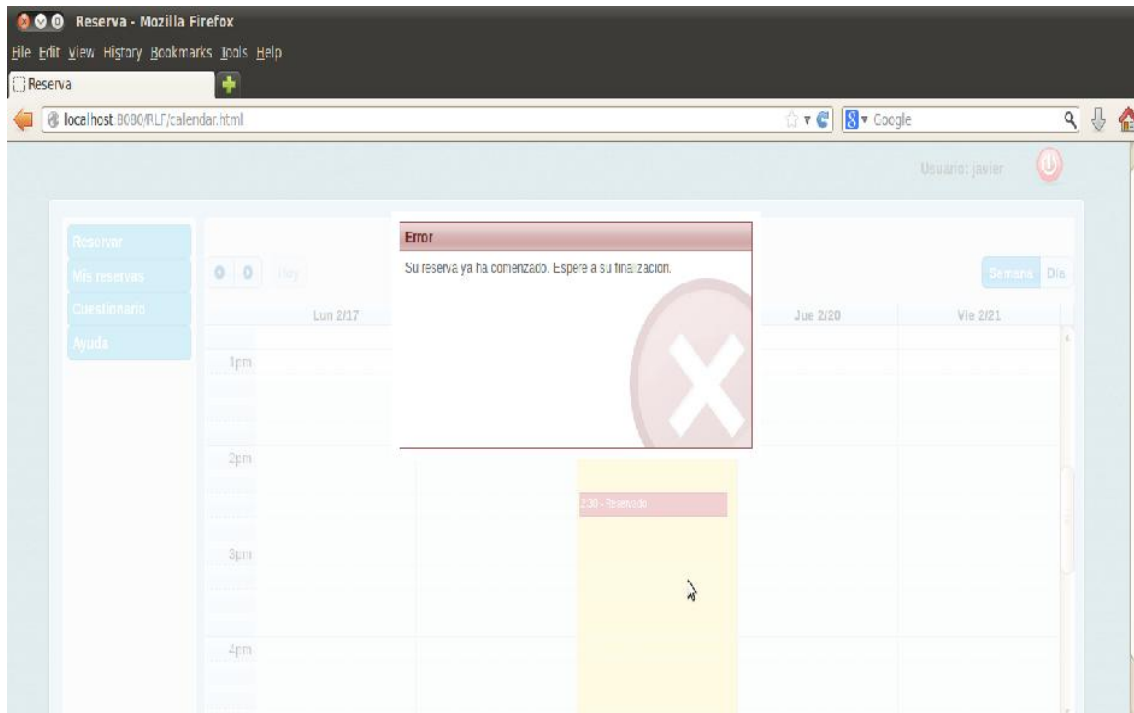


Ilustración 34: Intento de reservar teniendo una reserva comenzada

4.3.4. Ayuda

El usuario accederá a esta interfaz a través de la opción *Ayuda* del menú que se encuentra en cada página. Mostrará un menú de tipo acordeón en el que cada título, correspondiente a un tema de ayuda específico, desplegará una sección de texto correspondiente al desarrollo del tema.

Los siguientes temas de ayuda permitirán al usuario hacer un uso óptimo de los servicios de la aplicación:

- Realizar una reserva: se explicará cómo hacer una reserva y los significados de cada mensaje de la aplicación al respecto.
- Comenzar una reserva: se explicará cuándo se puede acceder a una reserva y cómo elegir los parámetros de configuración del PID.
- Salir de una reserva: se explicará el modo de parar las herramientas y salir correctamente de una reserva.
- Guardar los datos de una reserva: se explicará cómo y dónde se guardan los datos obtenidos tras la ejecución de las herramientas.

- Enviar un cuestionario: se explicará el uso de cuestionario.

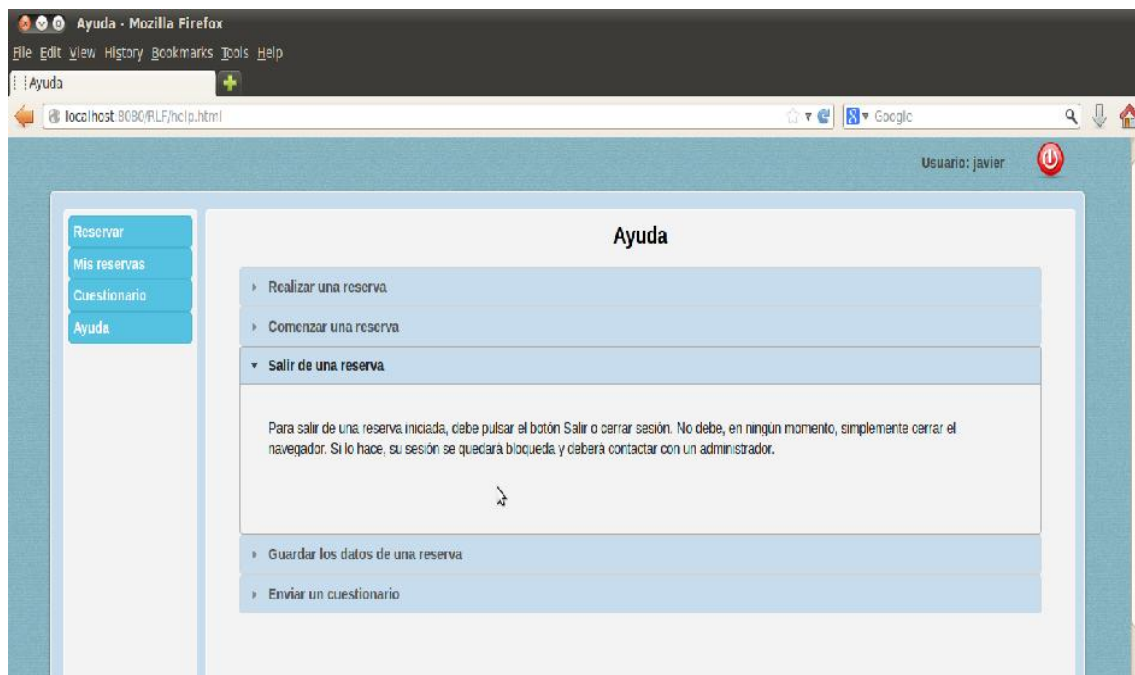


Ilustración 35: Ayuda

4.3.5. Cuestionario

El usuario accederá a esta interfaz a través de la opción *Cuestionario* del menú que se encuentra en cada página. Mostrará una lista de preguntas con varias respuestas, de las cuales el usuario podrá elegir solamente una. Cada pregunta tendrá marcada una respuesta por defecto, para evitar respuestas en blanco. Adicionalmente a la lista, al final de ésta se mostrará un botón *Enviar* que enviará las respuestas del usuario una vez pulsado. Este cuestionario podrá ser enviado tantas veces como el usuario crea necesario, pero solamente se guardará, para cada usuario, el último que se envíe.

El cuestionario servirá para conocer la opinión del usuario sobre la práctica realizada y elaborar estadísticas.

Cuestionario - Mozilla Firefox

File Edit View History Bookmarks Tools Help

localhost:8080/RLF/test.html

Usuario: javier

Cuestionario

Evalúe la dificultad personal que ha presentado la realización práctica de 1 a 5.

- ☒ 1 (Muy fácil)
- ☐ 2 (Fácil)
- ☐ 3 (Facilidad media)
- ☐ 4 (Difícil)
- ☐ 5 (Muy difícil)

Evalúe el funcionamiento de los recursos (vídeo y gráfica) utilizados durante la práctica de 1 a 5.

- ☒ 1 (Pésimo)
- ☐ 2 (Mal)
- ☐ 3 (Regular)
- ☐ 4 (Bueno)
- ☐ 5 (Excelente)

Enviar

Ilustración 36: Cuestionario

Cuestionario - Mozilla Firefox

File Edit View History Bookmarks Tools Help

localhost:8080/RLF/test.html

Usuario: javier

Cuestionario

Evalúe la dificultad personal que ha presentado la realización práctica de 1 a 5.

- ☒ 1 (Muy fácil)
- ☐ 2 (Fácil)
- ☐ 3 (Facilidad media)
- ☐ 4 (Difícil)
- ☐ 5 (Muy difícil)

Evalúe el funcionamiento de los recursos (vídeo y gráfica) utilizados durante la práctica de 1 a 5.

- ☒ 1 (Pésimo)
- ☐ 2 (Mal)
- ☐ 3 (Regular)
- ☐ 4 (Bueno)
- ☐ 5 (Excelente)

Enviar

Mensaje

El cuestionario ha sido enviado correctamente

Ilustración 37: Confirmación de envío del cuestionario

4.3.6. Configuración del motor

A esta interfaz se accederá cuando el usuario pulse el botón *Comenzar* de una reserva del listado mostrado por la interfaz *Listado de reservas*. Permitirá la elección de los parámetros de configuración y datos a enviar al PID, antes de comenzar la ejecución de las herramientas.

Esta página mostrará tres inputs de tipo *text*, uno de tipo *file* y un botón *Enviar*. Los inputs de tipo *text* permitirán introducir los valores Proporcional, Integral y Derivativo calculados previamente por el propio usuario. El input de tipo *file* permitirá elegir un fichero de datos elaborado por el usuario previamente también.

Los parámetros de configuración indicarán la configuración elegida para el PID, y el fichero de datos contendrá una lista de números que se le enviarán periódicamente a éste. Una vez rellenados los cuatro campos, el usuario podrá enviar sus elecciones. Se enviará al usuario a la interfaz *Carga de la reserva*.

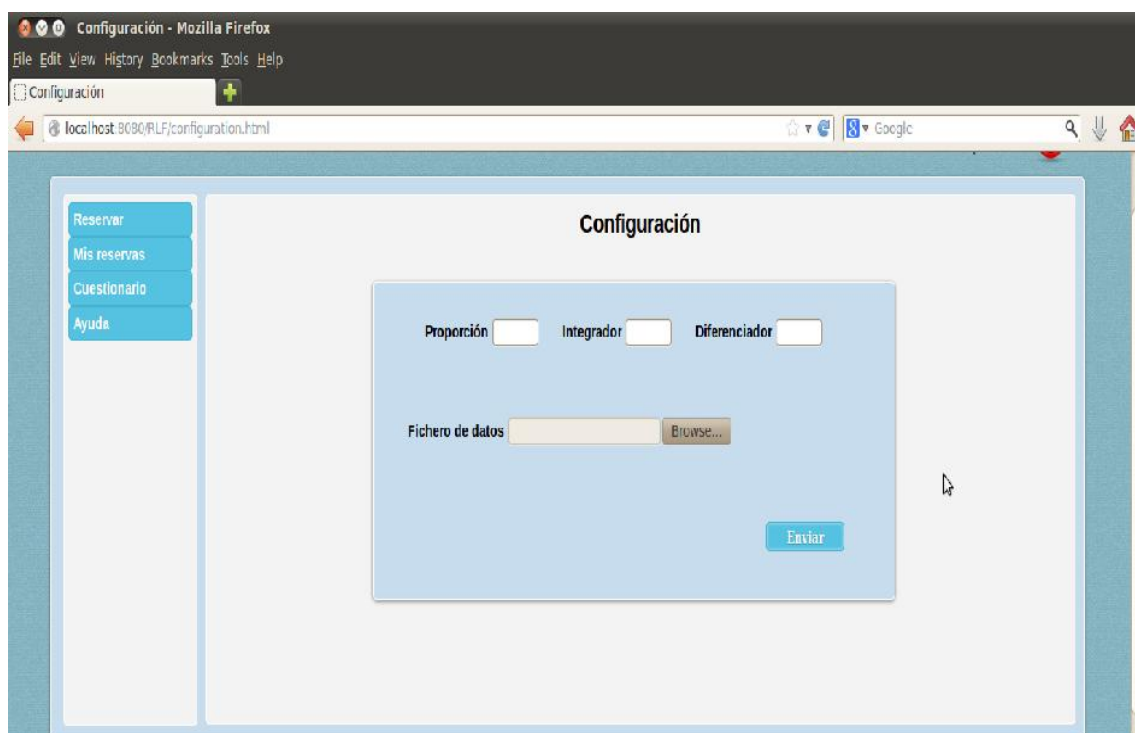


Ilustración 38: Configuración

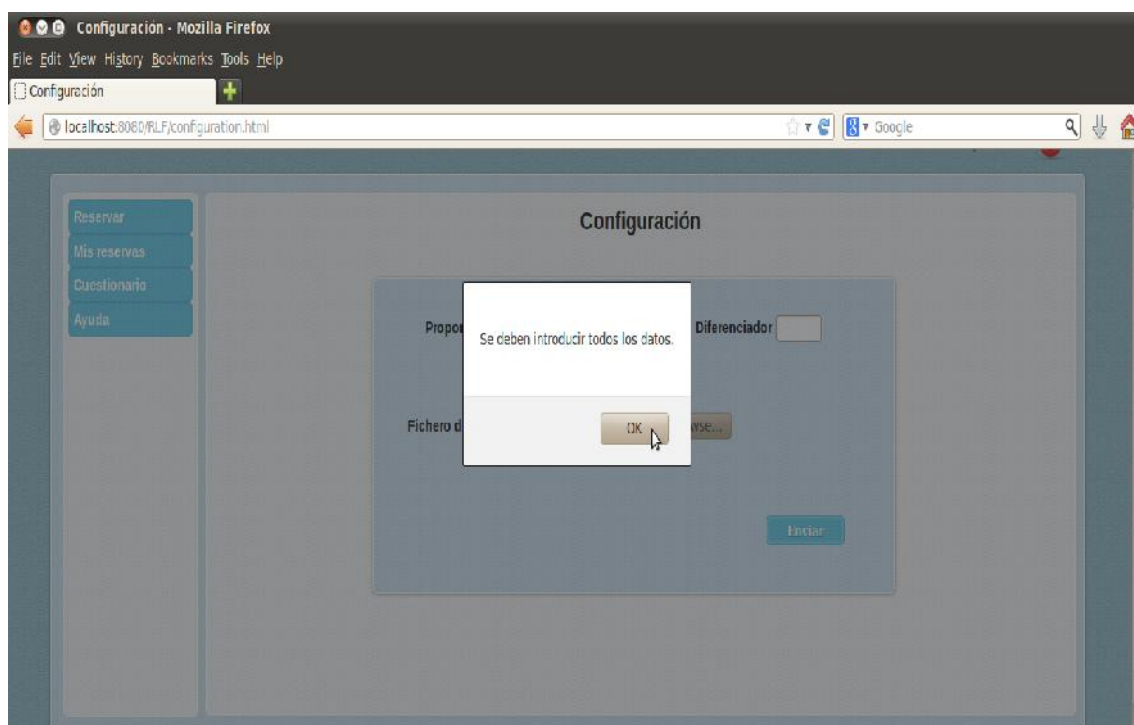


Ilustración 39: Datos no rellenados completamente

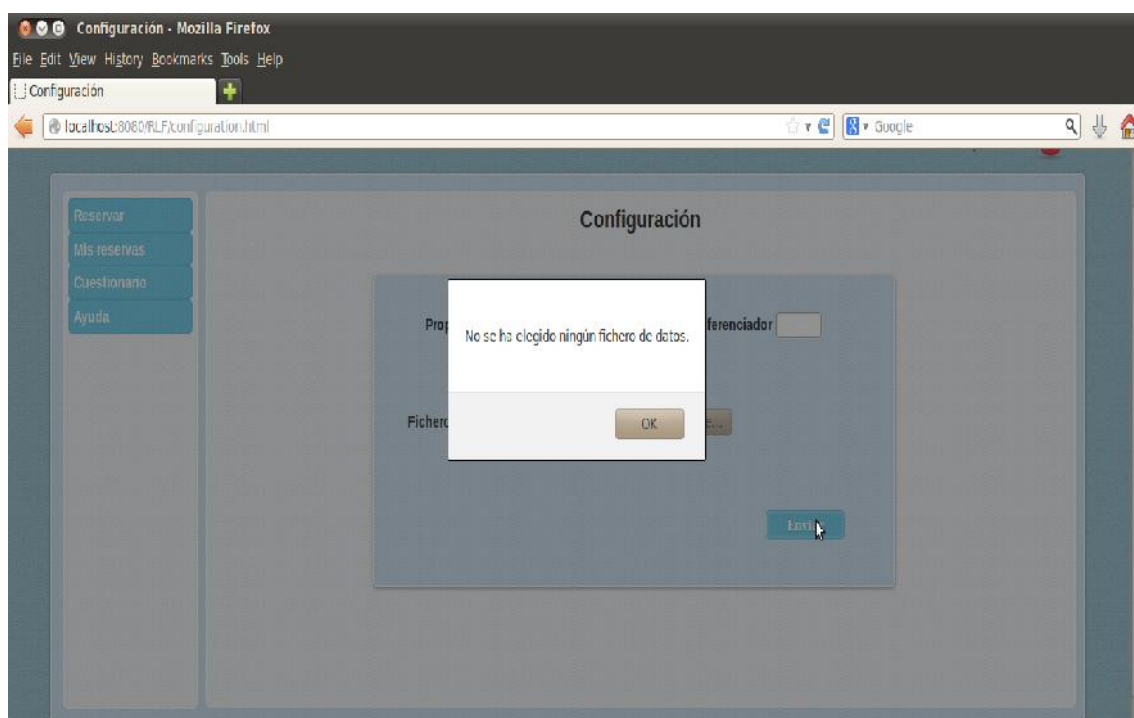


Ilustración 40: Fichero de datos no elegido

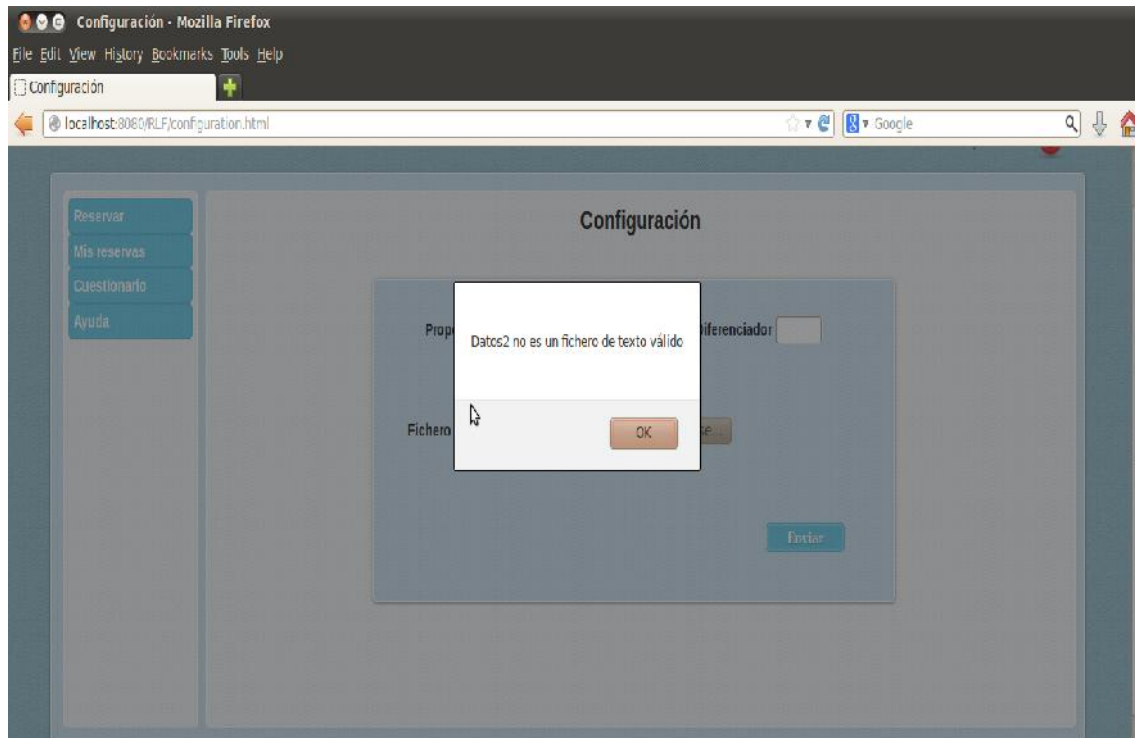


Ilustración 41: Fichero no válido

4.3.7. Carga de la reserva

A esta interfaz se accederá cuando el usuario pulse el botón *Enviar* de la interfaz *Configuración del motor*. Su objetivo es proporcionar una interfaz de espera para el usuario mientras se cargan las herramientas que usará la reserva de éste. Se mostrará una página con un fondo azul y un mensaje indicando que se está cargando. Una vez pase un tiempo predefinido (tiempo calculado en base a lo que tarda el video del motor en comenzar a transmitirse), el usuario será enviado a la interfaz *Ejecución de la reserva*.

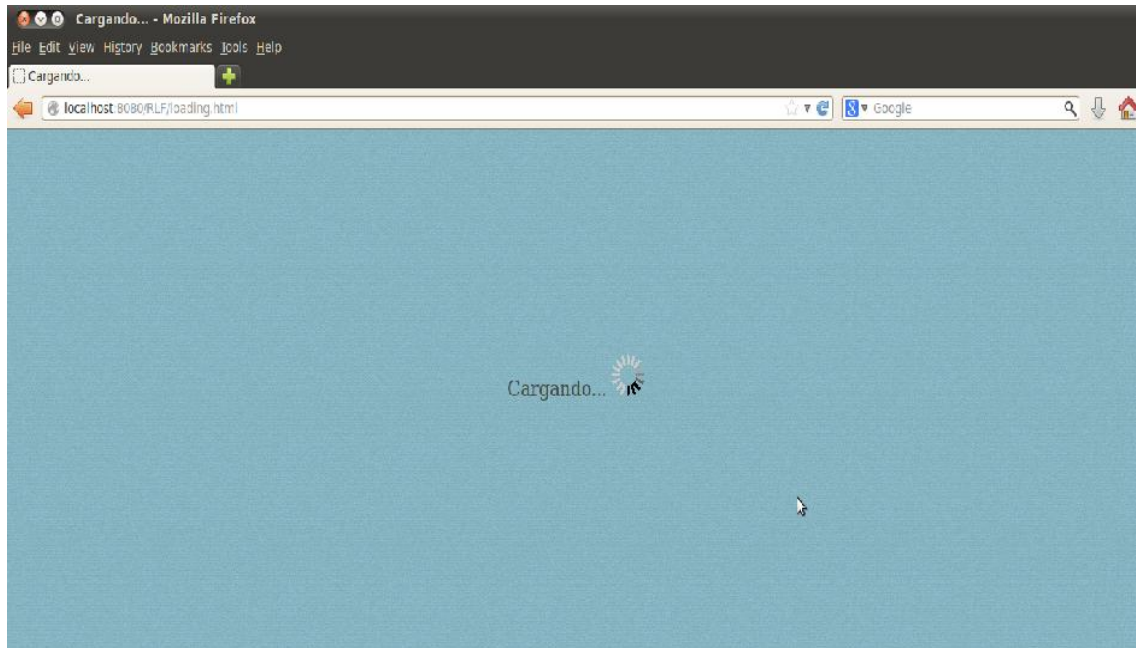


Ilustración 42: Página de espera

4.3.8. Ejecución de la reserva

A esta interfaz se accederá luego de pasado el tiempo de carga especificado por la interfaz *Carga de la reserva*. Mostrará el video del motor (determinado por los parámetros de configuración del PID y el fichero de datos enviados) y una gráfica detallando cada dato enviado al PID y la correspondiente respuesta**.

Los elementos presentes en la página serán los siguientes:

- **Video:** mostrará el video en tiempo real del motor y las reacciones provocadas en éste por los datos enviados (contenidos en el fichero de texto elegido previamente).
- **Botón *Comenzar* y datos:** el botón *Comenzar*, una vez pulsado, iniciará el envío de datos al PID. Cuando esto ocurra, el botón será eliminado y en la zona en la que se encontraba éste, se mostrará cada vez el valor del dato enviado y el dato recibido del PID.
- **Gráfica:** representará gráficamente en el tiempo dos líneas de datos. La línea azul representará los datos enviados al PID y la amarilla las repuestas del PID correspondientes a cada dato enviado.

- **Cuenta regresiva:** mostrará el tiempo restante para la finalización de la reserva. Una vez llegue a 0, se detendrá la ejecución de las herramientas, se preguntará al usuario si desea guardar los datos e, independientemente de la respuesta, se le enviará a la interfaz *Listado de reservas*, que mostrará el listado actualizado al haber sido borrada la reserva de la BBDD.
- **Botón Salir:** brindará un mecanismo de salida de la reserva por el usuario. Una vez pulsado, se parará la ejecución de las herramientas y se permitirá al usuario decidir si desea guardar los datos enviados y recibidos durante la ejecución (pop-up). Cualquiera sea la decisión del usuario, éste será enviado a la interfaz *Listado de reservas*, desde la cual podrá volver a la reserva si así lo deseara.

El menú no estará presente en esta página, para garantizar la parada correcta de las herramientas y permitir que el usuario pueda acceder a la reserva nuevamente sin ningún problema derivado de una salida errónea.

Los siguientes pantallazos muestran la interfaz de ejecución sin el video, porque los pantallazos no lograron captarlo correctamente:

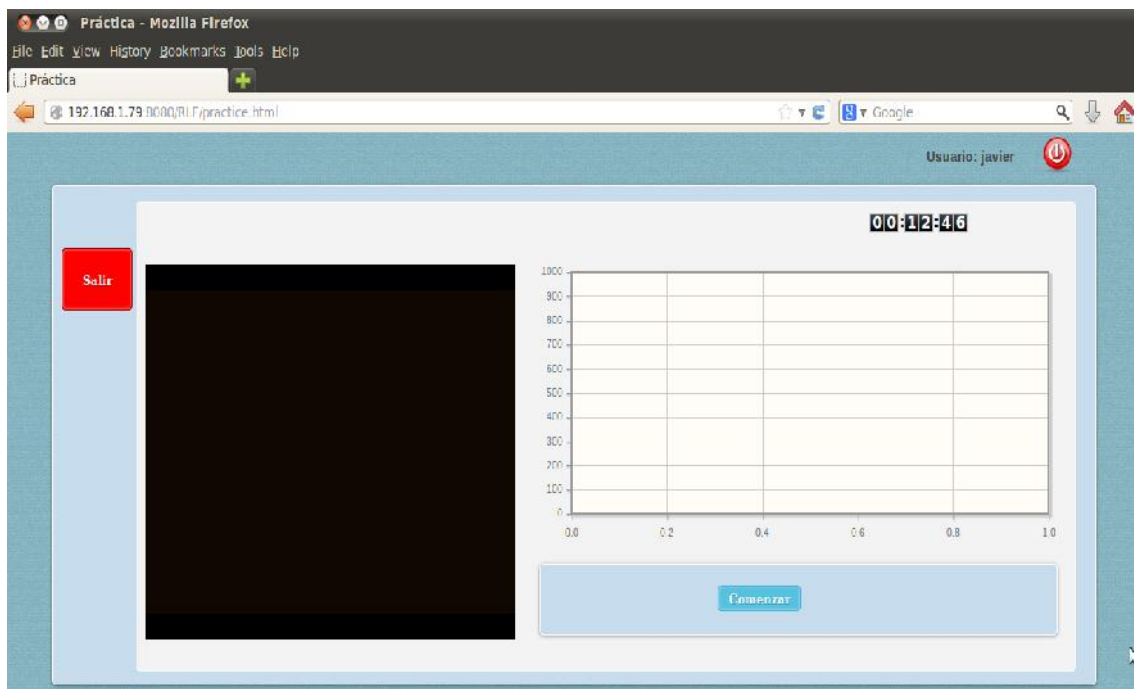


Ilustración 43: Estado inicial de la página de ejecución (pantallazo)

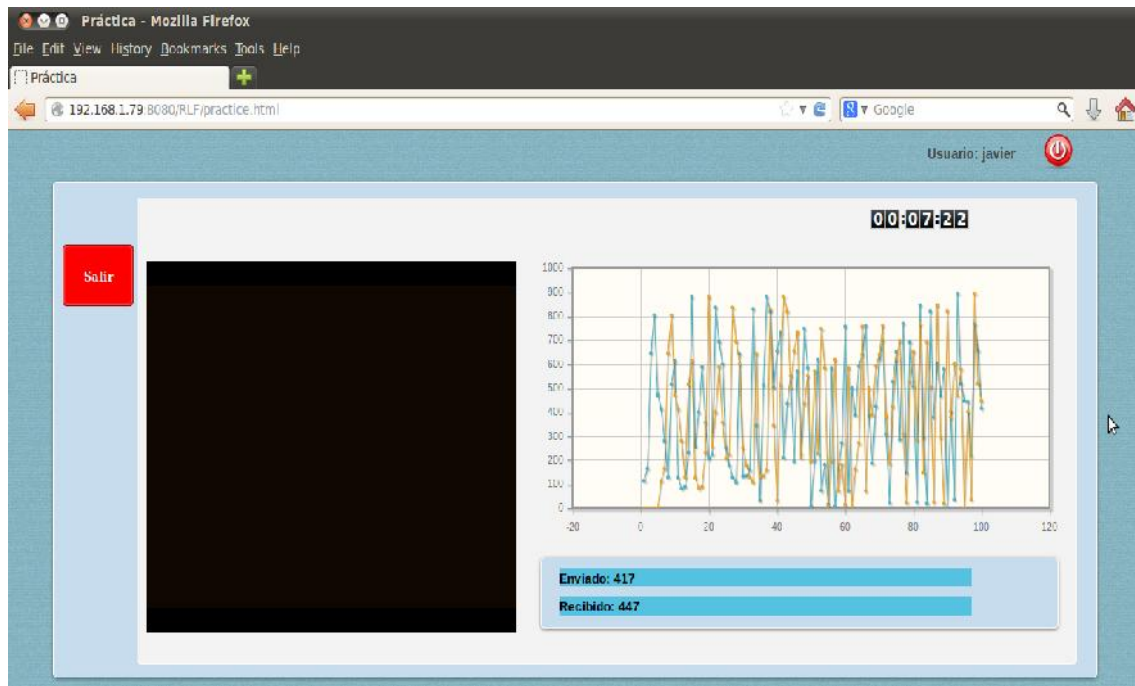


Ilustración 44: Ejecución (pantallazo)

Las siguientes imágenes son fotos de la interfaz, en las que sí se puede apreciar el video:

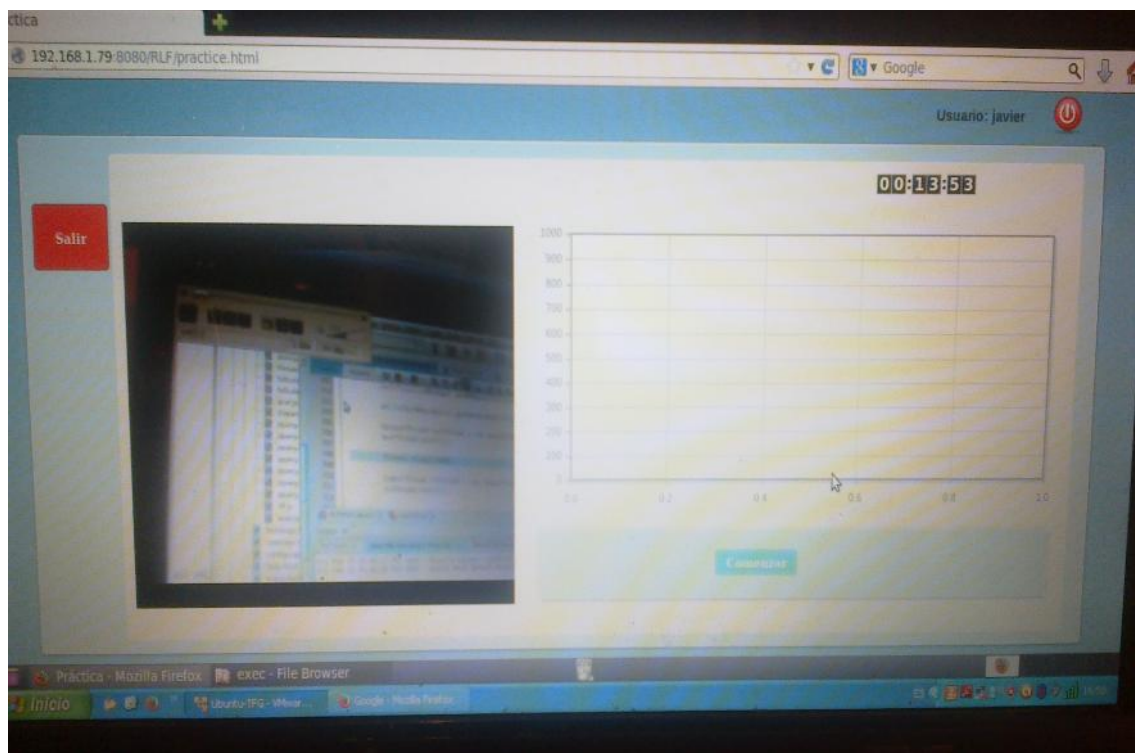


Ilustración 45: Estado inicial de la página de ejecución (foto)

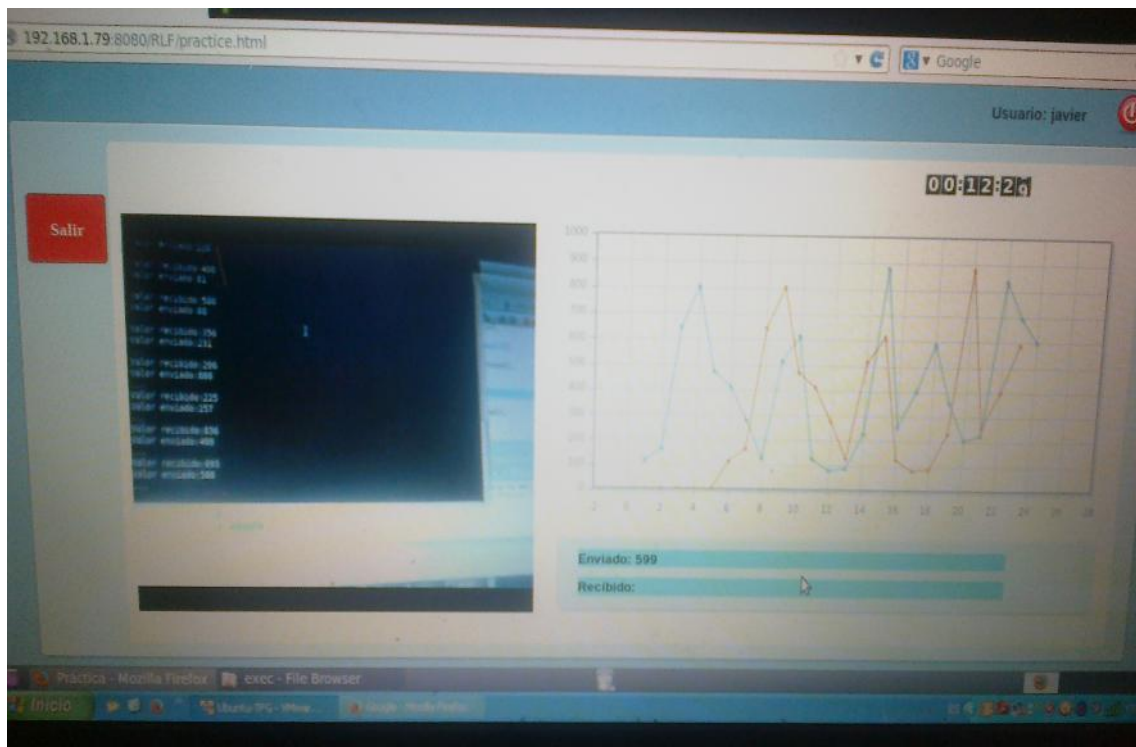


Ilustración 46: Ejecución (foto)

A continuación se muestra en imágenes el proceso de guardado:

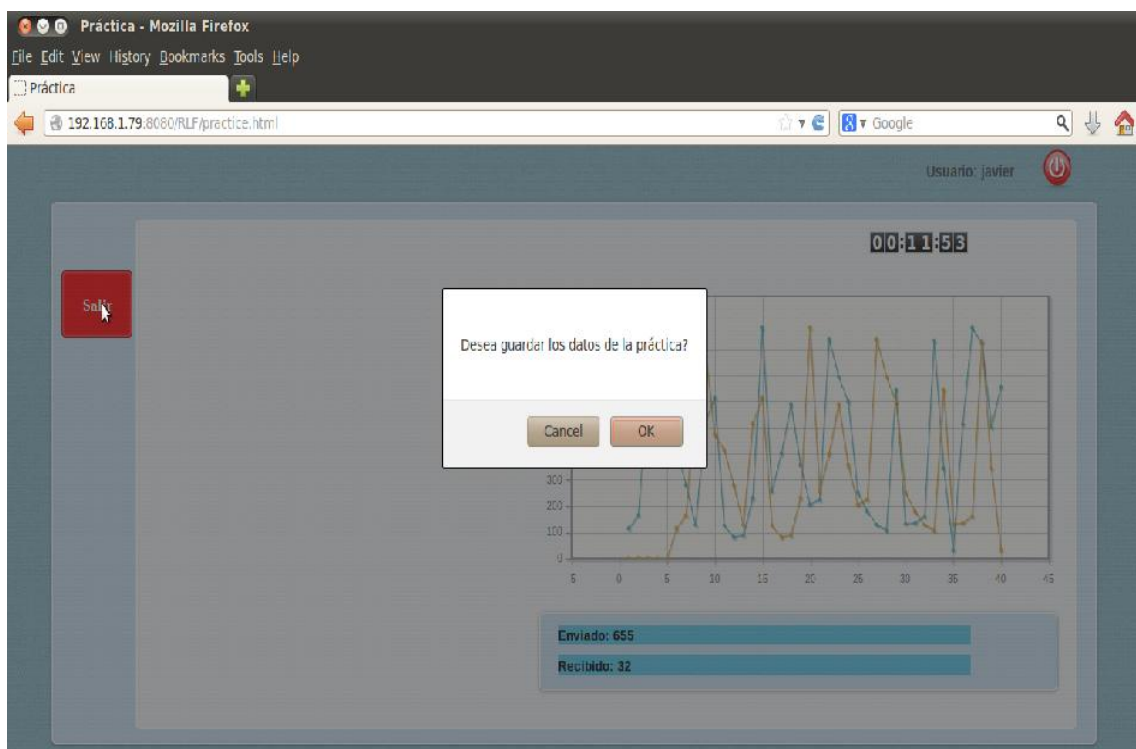


Ilustración 47: Opción de guardado

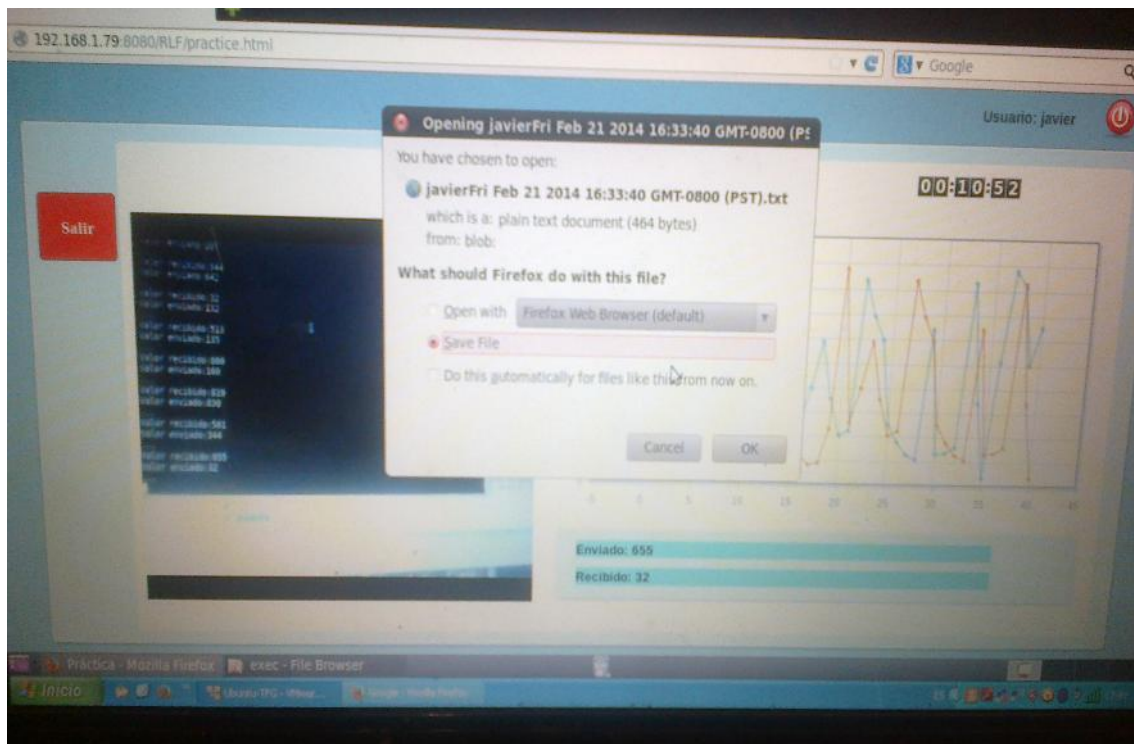


Ilustración 48: Opción de abrir o guardar el fichero

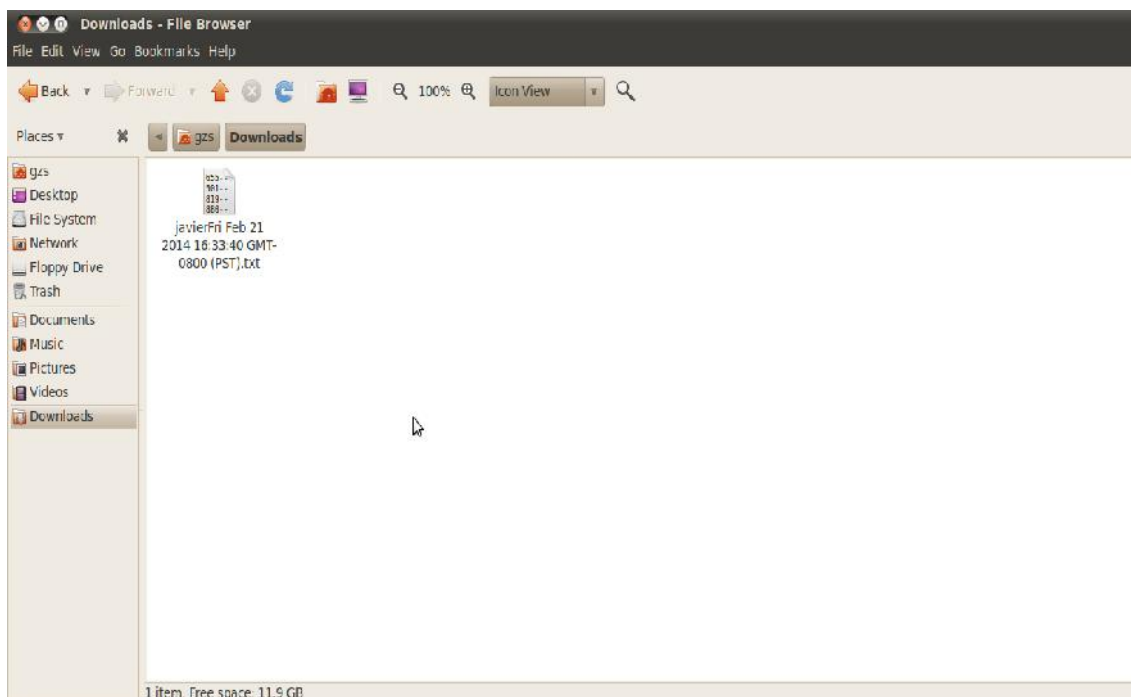
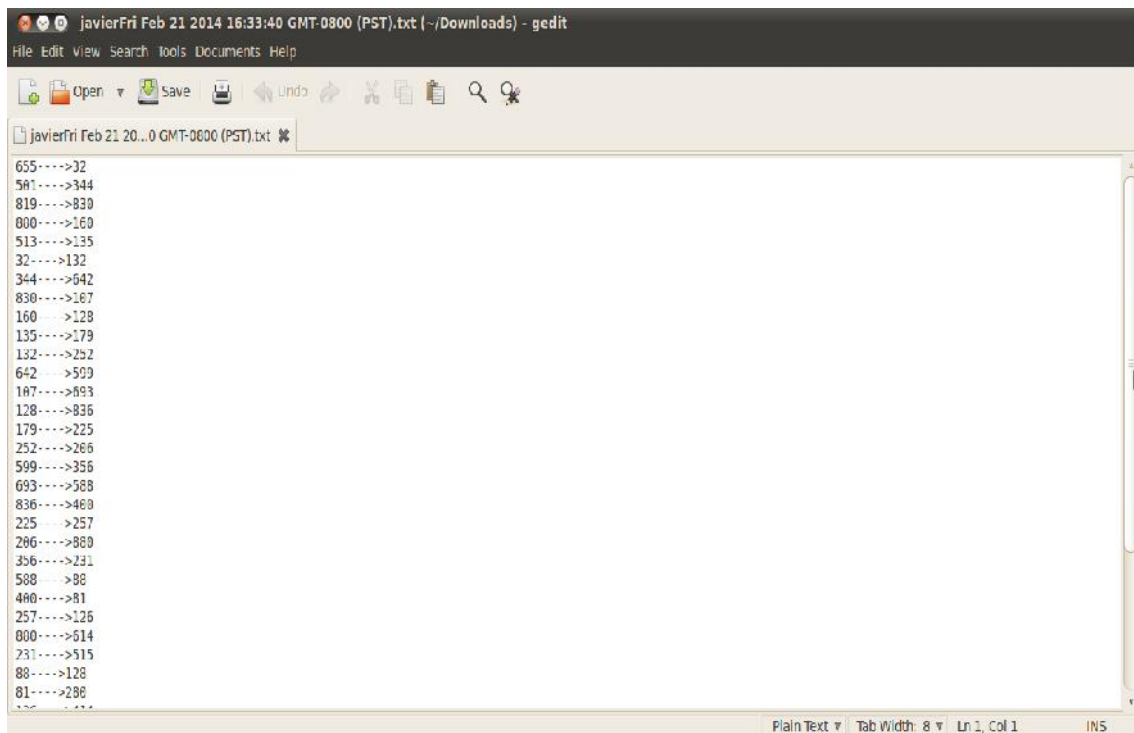


Ilustración 49: Fichero creado



```
javierFri Feb 21 2014 16:33:40 GMT-0800 (PST).txt
655---->32
581---->344
819---->830
000---->160
513---->135
32---->132
344---->642
830---->167
160---->128
135---->179
132---->252
642---->599
187---->893
128---->836
179---->225
252---->266
599---->356
693---->588
836---->460
225---->257
286---->880
356---->231
588---->88
480---->81
257---->126
000---->614
731---->515
88---->128
81---->280
```

Ilustración 50: Contenido del fichero

****** Como se ha aclarado anteriormente, la comunicación entre la herramienta y el motor aún no está implementada, por lo que se realizará una simulación. Los detalles de la simulación serán explicados en el apartado 5.2. Herramientas.

4.4. Diagrama de navegación

El siguiente diagrama de navegación mostrará el orden y la posibilidad de acceso a las interfaces de la aplicación:

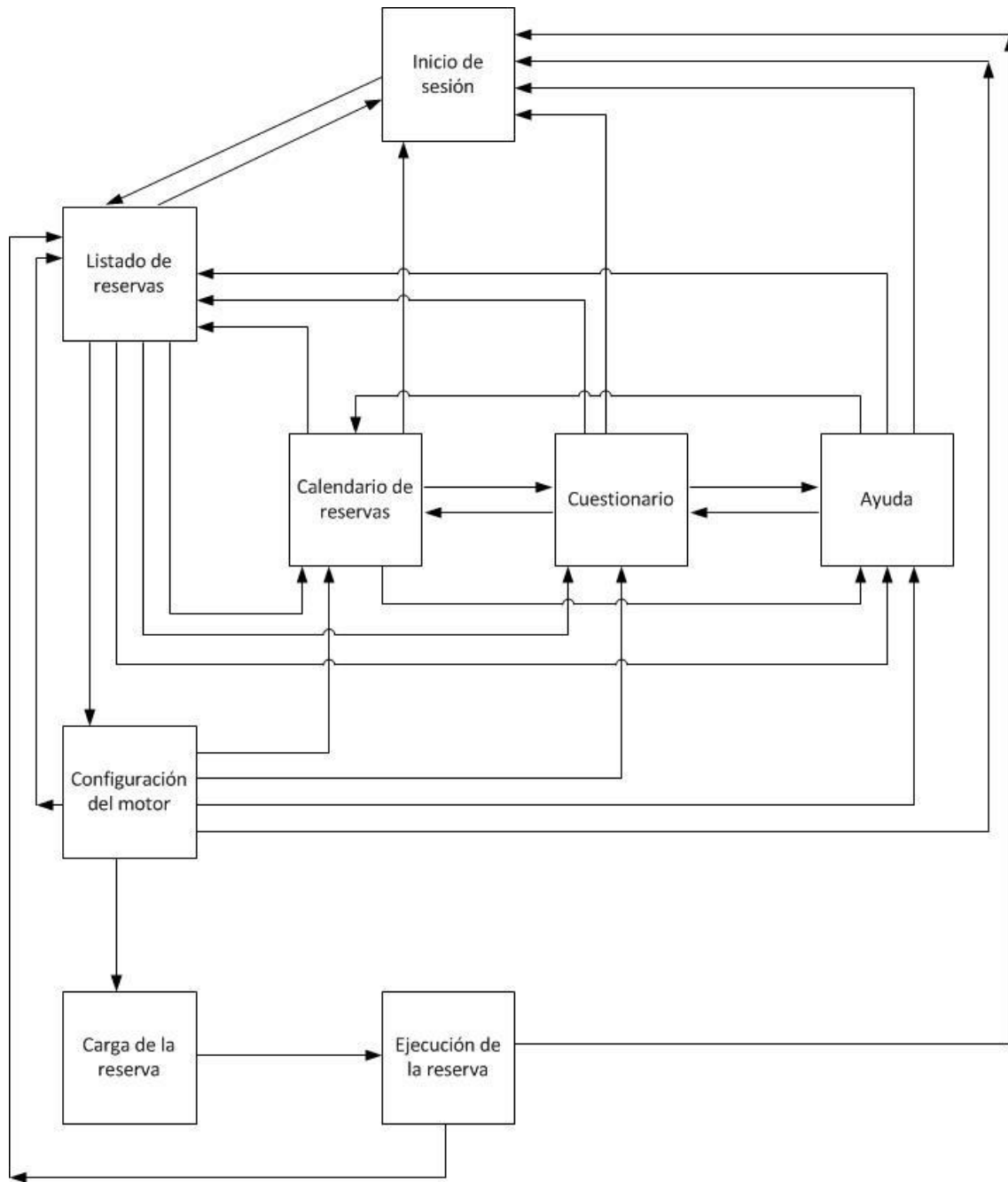


Ilustración 51: Diagrama de navegación

Del diagrama se deduce la siguiente información sobre cada interfaz:

- **Inicio de sesión**: desde esta interfaz solo se puede acceder a la interfaz *Listado de reservas*. Esto ocurre cuando el usuario inicia sesión.

- **Listado de reservas:** desde esta interfaz, debido al menú que está presente en cada página (excepto la página de ejecución de la reserva), se puede acceder a las interfaces *Calendario de reservas*, *Cuestionario* y *Ayuda*. Por otro lado, se accede a *Configuración del motor* si el usuario decide comenzar la reserva. Finalmente, se puede volver a la interfaz *Inicio de sesión* si se cierra sesión.
- **Calendario de reservas:** desde esta interfaz, también debido al menú, se puede acceder a las interfaces *Listado de reservas*, *Cuestionario* y *Ayuda*. Se puede volver a la interfaz *Inicio de sesión* si se cierra sesión.
- **Cuestionario:** desde esta interfaz, debido al menú, se puede acceder a las interfaces *Listado de reservas*, *Calendario de reservas* y *Ayuda*. Se puede volver a la interfaz *Inicio de sesión* si se cierra sesión.
- **Ayuda:** desde esta interfaz, debido al menú, se puede acceder a las interfaces *Listado de reservas*, *Calendario de reservas* y *Cuestionario*. Se puede volver a la interfaz *Inicio de sesión* si se cierra sesión.
- **Configuración del motor:** desde esta interfaz, debido al menú, se puede acceder a las interfaces *Listado de reservas*, *Calendario de reservas*, *Cuestionario* y *Ayuda*. Se puede volver a la interfaz *Inicio de sesión* si se cierra sesión. Finalmente, tras enviar los datos de configuración del PID, se puede pasar a la interfaz *Carga de la reserva*.
- **Carga de la reserva:** desde esta interfaz solo se puede acceder a la interfaz *Ejecución de la reserva*. Esto ocurre una vez el tiempo de carga definido ha pasado.
- **Ejecución de la reserva:** en esta interfaz no existe menú. Se puede volver a la interfaz *Inicio de sesión* si se cierra sesión. Una vez terminada la reserva o habiendo el usuario decidido salir de la reserva, se pasa a la interfaz *Listado de reservas*.

5. Implementación

En este capítulo se detallará la implementación del proyecto. Se especificarán los métodos de comunicación cliente-servidor y el funcionamiento de las herramientas creadas.

5.1. Servidor

El servidor se encargará de atender todas las peticiones de los clientes, que pueden ser tanto consultas a bases de datos, como lectura de ficheros y comunicación con los laboratorios remotos. Las peticiones del cliente serán enviadas al servidor dentro de elementos JSON, cuyos campos dependerán de la petición que el cliente realice.

En el lado del servidor se creará un WebSocket Endpoint cuya dirección se utilizará para crear el WebSocket del lado del cliente. De este modo se manejará la comunicación por WebSockets. Por otro lado, el servidor estará conectado a la base de datos de la aplicación.

En el evento `onMessage` del WebSocket Endpoint del servidor se atenderán las peticiones del cliente. Las peticiones que atenderá el servidor serán las siguientes (identificadas por el nombre de operación):

➤ login

Esta petición ocurre cuando el usuario intenta iniciar sesión y su fin es comprobar la autenticidad de los datos de inicio de sesión.

El cliente envía al servidor (utilizando la operación *send* del WebSocket) un objeto JSON con los campos *operation*, *user* y *password*. Estos campos tendrán los valores:

- *operation*: "login"
- *user*: usuario introducido para iniciar sesión
- *password*: contraseña introducida para iniciar sesión

El servidor recibe la petición del usuario y, en base a la operación contenida en el mensaje, ejecuta una serie de pasos. Para la operación *login*, el servidor hace una consulta a la base de datos seleccionando el usuario cuyo nombre de usuario y contraseña coincidan con los enviados por el cliente.

El servidor envía a través del WebSocket un JSON conteniendo los campos *operation* con valor "login" y *response*. El valor de *response* dependerá del resultado de la consulta. Si el usuario se encuentra en la base de datos,

response tendrá el valor “Y”. De lo contrario, el valor será “N”. Adicionalmente, en el caso de ser “Y”, se añadirá al JSON los campos *token*, *tool*, *user*, *password* y *booking_time*.

- *token*: token creado por un cifrador a partir del nombre de usuario y contraseña. Se utilizará para las operaciones con el laboratorio remoto.
- *tool*: id de la herramienta reservada (solo una, RLF_Chart)
- *user*: usuario introducido para iniciar sesión
- *password*: contraseña introducida para iniciar sesión
- *booking_time*: tiempo de duración de la reserva (valor predefinido)

En el evento *onMessage* del WebSocket del lado del cliente, si el mensaje recibido del servidor contiene la operación *login* y el valor de *response* es “N”, se le muestra al usuario un mensaje de error y no se inicia sesión.

Contrariamente, si el valor de *response* es “Y”, el cliente guarda el usuario, la contraseña, el token, la herramienta y el tiempo de reserva enviado por el servidor en *sessionStorage* y se pasa a la página del listado de reservas (se ha iniciado sesión).

sessionStorage es un servicio que brinda HTML5 Web Storage. Permite guardar información de la sesión localmente dentro del browser. Posibilita almacenar grandes cantidades de información sin afectar el rendimiento de la aplicación. Por otro lado, la información solamente se usa si es necesitada y no se transfiere al servidor.

➤ **getHelpThemes**

Esta petición ocurre cuando el usuario accede a la página de ayuda y su objetivo es obtener los contenidos de ayuda para cargarlos en la página.

El cliente envía al servidor un objeto JSON con el campo *operation*.

- *operation*: "getHelpThemes"

El servidor recibe la petición del usuario y, para la operación *getHelpThemes*, el servidor lee el fichero predefinido en el que se encuentran los contenidos de ayuda. Añade cada título de la ayuda y su contenido a una estructura JSON, bajo los nombres de *title* y *content*, respectivamente. Añade, así mismo, el campo *operation* con valor “getHelpThemes” y *response* con valor “Y”.

En el lado del cliente, si el mensaje recibido del servidor contiene la operación *getHelpThemes*, el cliente se encargará de añadir dinámicamente a la página de ayuda los títulos y contenidos enviados por el servidor.

➤ **getQuestions**

Esta petición ocurre cuando el usuario accede a la página del cuestionario y su finalidad es obtener el listado de preguntas y respuestas del cuestionario para cargarlos en la página.

El cliente envía al servidor un objeto JSON con el campo *operation*.

- *operation*: "getQuestions"

El servidor recibe la petición del usuario y, para la operación *getQuestions*, el servidor lee el fichero predefinido en el que se encuentran las preguntas y sus posibles respuestas. Añade cada pregunta y cada una de sus respuestas a una estructura JSON, bajo los nombres de *question* y *answer*, respectivamente. Añade, así mismo, el campo *operation* con valor “getQuestions” y *response* con valor “Y”.

En el lado del cliente, si el mensaje recibido del servidor contiene la operación *getQuestions*, el cliente se encargará de añadir dinámicamente a la página del cuestionario las preguntas y respuestas enviadas por el servidor.

➤ **storeTest**

Esta petición ocurre cuando el usuario envía al cuestionario relleno y su fin es guardar las respuestas dadas por el usuario al cuestionario.

El cliente envía al servidor un objeto JSON con los campos *operation*, *user* y las respuestas a cada pregunta a través de los campos *question* y *answer*. Estos campos tendrán los valores:

- *operation*: "storeTest"
- *user*: el usuario guardado en sessionStorage tras el inicio de sesión
- *question+índice*: pregunta
- *answer+índice*: respuesta

El servidor recibe la petición del usuario y, para la operación *storeTest*, hace una consulta a la base de datos insertando (o actualizando, en caso de existir un cuestionario previo del mismo usuario) en la tabla *test* las respuestas dadas por el usuario a las preguntas.

Esta petición no tiene respuesta para el cliente.

➤ **getBookings**

Esta petición ocurre cuando el usuario accede a la página del calendario para realizar una reserva. Su objetivo es obtener las reservas que existen para cargarlas en el calendario.

El cliente envía al servidor un objeto JSON con los campos *operation* y *user*. Estos campos tendrán los valores:

- *operation*: "getBookings"
- *user*: el usuario guardado en sessionStorage tras el inicio de sesión

El servidor recibe la petición del usuario y, para la operación *getBookings*, hace una consulta a la base de datos seleccionando la fecha y usuario de todas las reservas.

El servidor envía a través del WebSocket un JSON conteniendo los campos *operation* con valor "getBookings" y *response*. El valor de *response* dependerá del resultado de la consulta. Si existen reservas que cargar, *response* tendrá el valor "Y". De lo contrario, el valor será "N". Adicionalmente, en el caso de ser

“Y”, se añadirá al JSON los campos *year, month, day, hours, minutes, user* y *color*.

- *year, month, day, hours, minutes*: campos correspondientes a la fecha de cada reserva.
- *user*: usuario que ha hecho la reserva
- *color*: color que tendrá la franja de tiempo correspondiente a la reserva (verde si es una reserva del propio usuario que ha iniciado sesión, roja si no lo es o gris si, en cualquier caso, la reserva ya ha comenzado)

En el evento *onMessage* del WebSocket del lado del cliente, si el mensaje recibido del servidor contiene la operación *getBookings* y el valor de *response* es “Y”, el cliente se encargará de añadir dinámicamente al calendario las reservas enviadas por el servidor.

➤ **storeBooking**

Esta petición ocurre cuando el usuario hace una reserva para una fecha específica. Su finalidad es guardar dicha reserva.

El cliente envía al servidor un objeto JSON con los campos *operation, user, year, month, day, hours* y *minutes*. Estos campos tendrán los valores:

- *operation*: "storeBooking"
- *user*: el usuario guardado en *sessionStorage* tras el inicio de sesión
- *year, month, day, hours, minutes*: campos correspondientes a la fecha elegida por el usuario para la reserva.

El servidor recibe la petición del usuario y, para la operación *storeBooking*, hace una consulta a la base de datos insertando (o actualizando, en caso de existir una reserva previa del mismo usuario) en la tabla *booking* la reserva hecha por el usuario. Dado que esta aplicación solamente utilizará dos herramientas y ambas se deben usar en la reserva como si fuesen una sola, se harán dos

inserciones de reserva, una para cada herramienta. Se utilizarán directamente los id's de las herramientas que por el momento se tienen en la aplicación.

Esta petición no tiene respuesta para el cliente.

➤ **logout**

Esta petición ocurre cuando el usuario cierra sesión y su fin es detener todos los procesos que estén iniciados por el cliente.

El cliente envía al servidor (utilizando la operación *send* del WebSocket) un objeto JSON con los campos *operation* y *token*. Estos campos tendrán los valores:

- *operation*: "logout"
- *token*: token asignado al usuario tras el inicio de sesión

El servidor recibe la petición del usuario y, para la operación *logout*, el servidor cierra los sockets (si los hay) que pueda estar utilizando el usuario (si estuviese utilizando una herramienta), se notifica a los laboratorios y se elimina el token.

Esta petición no tiene respuesta para el cliente.

➤ **getUserBookings**

Esta petición ocurre cuando el usuario accede a la página con el listado de sus reservas. Su objetivo es obtener los datos de las reservas que existen del usuario actual para cargarlas en el listado.

El cliente envía al servidor un objeto JSON con los campos *operation* y *user*. Estos campos tendrán los valores:

- *operation*: "getUserBookings"
- *user*: el usuario guardado en sessionStorage tras el inicio de sesión

El servidor recibe la petición del usuario y, para la operación *getUserBookings*, hace una consulta a la base de datos seleccionando la fecha de todas las reservas realizadas por el usuario.

El servidor envía a través del WebSocket un JSON conteniendo los campos *operation* con valor "getUserBookings" y *response*. El valor de *response* dependerá del resultado de la consulta. Si el usuario tiene alguna reserva, *response* tendrá el valor "Y". De lo contrario, el valor será "N". Adicionalmente, en el caso de ser "Y", se añadirá al JSON los campos *year*, *month*, *day*, *hours* y *minutes*.

- *year*, *month*, *day*, *hours*, *minutes*: campos correspondientes a la fecha de cada reserva del usuario.

En el evento *onMessage* del WebSocket del lado del cliente, si el mensaje recibido del servidor contiene la operación *getUserBookings* y el valor de *response* es "Y", el cliente se encargará de añadir dinámicamente al listado los datos de las reservas del usuario enviadas por el servidor. Si *response* es "N", el listado estará vacío, conteniendo una sola fila con un mensaje indicativo de la ausencia de reservas.

➤ **getTimer**

Esta petición ocurre cuando el usuario accede a la página de ejecución de la reserva. Su fin es obtener la fecha de la reserva actual, para mostrar al usuario una cuenta regresiva con el tiempo restante de la reserva (que se calcula en base a la fecha de inicio de la reserva y la hora en el momento de acceder).

El cliente envía al servidor un objeto JSON con los campos *operation* y *user*. Estos campos tendrán los valores:

- *operation*: "getTimer"
- *user*: el usuario guardado en *sessionStorage* tras el inicio de sesión

El servidor recibe la petición del usuario y, para la operación *getTimer*, hace una consulta a la base de datos seleccionando la fecha de la reserva realizada por el usuario (la consulta se hace sobre todas las reservas, pero la restricción de que el usuario solamente puede tener una reserva de una misma herramienta, garantiza que solo habrá un resultado).

El servidor envía a través del WebSocket un JSON conteniendo los campos *operation* con valor “getTimer”, *year*, *month*, *day*, *hours* y *minutes*.

- *year*, *month*, *day*, *hours*, *minutes*: campos correspondientes a la fecha de la reserva.

En el evento *onMessage* del WebSocket del lado del cliente, si el mensaje recibido del servidor contiene la operación *getTimer*, el cliente se encargará de calcular el tiempo restante a partir de la fecha enviada por el servidor y crear la cuenta regresiva.

➤ **clearBookings**

Esta petición ocurre cuando la reserva del usuario se debe eliminar de la base de datos, ya sea porque la reserva ha terminado o porque el usuario decida cancelarla.

El cliente envía al servidor un objeto JSON con los campos *operation* y *user*. Estos campos tendrán los valores:

- *operation*: "clearBookings"
- *user*: el usuario guardado en *sessionStorage* tras el inicio de sesión

El servidor recibe la petición del usuario y, para la operación *clearBookings*, hace una consulta a la base de datos eliminando de la tabla *booking* las reservas del usuario. En un futuro, si se incluyen más herramientas y se admiten varias reservas, esto habría de modificarse para que solo eliminase la reserva correspondiente a herramientas específicas.

Esta petición no tiene respuesta para el cliente.

➤ **saveDataToFile**

Esta petición ocurre si el usuario decide guardar los datos resultantes de la ejecución de las herramientas (al finalizar la reserva, o si el usuario decide salir de ésta). Su finalidad es guardar en un fichero los datos de dicha ejecución.

El cliente envía al servidor un objeto JSON con los campos *operation* y *user*. Estos campos tendrán los valores:

- *operation*: "saveDataToFile"
- *user*: el usuario guardado en *sessionStorage* tras el inicio de sesión

El servidor recibe la petición del usuario y, para la operación *saveDataToFile*, envía a través del WebSocket un JSON conteniendo los campos *operation* con valor "saveDataToFile" y los pares de datos (enviado y recibido).

Esta información se sacará de la base de datos que, como se explicará, almacenará los datos de la última ejecución de la reserva del usuario.

En el lado del cliente, se creará un fichero conteniendo los datos enviados, que el usuario podrá abrir o guardar en la ruta que elija.

➤ **getCam**

Esta petición se envía al servidor una vez el usuario ha introducido y enviado los parámetros de configuración del PID y el fichero de datos que desea usar para la ejecución de su reserva. Su objetivo es preparar las herramientas para su uso y comenzar la transmisión de video del motor.

El cliente envía al servidor un objeto JSON con los campos *operation*, *token* y *user*. Estos campos tendrán los valores:

- *operation*: "getCam"
- *user*: el usuario guardado en *sessionStorage* tras el inicio de sesión
- *token*: token asignado al usuario tras el inicio de sesión

El servidor recibe la petición del usuario y, para la operación *getCam*, lo primero que hace es borrar de la base de datos todos los pares de datos resultantes de reservas antiguas del usuario. De este modo, solamente se guardarán los datos de la nueva ejecución (si al usuario así lo decide).

Seguidamente se envía el token del usuario al laboratorio en el que se encuentran las herramientas, para que verifique su autenticidad y se ejecuta el hilo que se encargará de gestionar la conversación a través de sockets entre el servidor y el laboratorio. Este hilo obtiene el socket destinado a la notificación (predefinido) del laboratorio y se dedica a escuchar las respuestas del

laboratorio. Si el servidor envía una petición de ejecución al laboratorio, este hilo se encarga de enviar esta petición y recibir la respuesta. Si ésta es afirmativa por parte del laboratorio (todas la comprobaciones hechas por el laboratorio han sido favorables), se crean los sockets de entrada y salida de datos que se usarán para la comunicación directa entre la herramienta y el servidor.

En el caso de *getCam*, se envía al laboratorio una petición de ejecución de la herramienta *RLF_Cam*, con la acción *stream* y se crea un hilo para leer del socket de salida (por el que llegan las comunicaciones de la herramienta). Cuando se ejecuta la herramienta (se ha lanzado el VLC Player) y llega un mensaje de confirmación por el socket, el servidor envía a través del WebSocket un JSON conteniendo los campos *operation* con valor “getCam” y *response* con valor “Y”.

En el lado del cliente, al recibir el mensaje, se envía al usuario a la página de espera, para garantizar que el VLC Player haya tenido tiempo de cargarse adecuadamente y comenzar a transmitir. Una vez que haya pasado el tiempo predefinido, se envía al usuario a la página de la ejecución, donde se visualizará el video resultante de la ejecución de la herramienta *RLF_Cam*.

➤ **sendData**

Esta petición se envía al servidor una vez que el usuario ha pulsado el botón Comenzar de la página de ejecución. Su finalidad es iniciar el envío de datos al PID, utilizando los datos contenidos en el fichero elegido por el usuario.

El cliente envía al servidor un objeto JSON con los campos *operation*, *proportional*, *integral*, *derivative*, *datafile* y *user*. Estos campos tendrán los valores:

- *operation*: "sendData"
- *user*: el usuario guardado en sessionStorage tras el inicio de sesión
- *datafile*: fichero de datos elegido por el usuario

- *proportional*: parámetro introducido por el usuario
- *integral*: parámetro introducido por el usuario
- *derivative*: parámetro introducido por el usuario

Dado que las herramientas ya están listas para su ejecución tras la petición *getCam*, el servidor envía al laboratorio una petición de ejecución de la herramienta *RLF_Chart*, con la acción *send* y los parámetros *proportional*, *integral* y *derivative*. Se crea un hilo para leer del socket de salida (por el que llegarán las comunicaciones de la herramienta) y otro para enviar por el socket de entrada (por el que se enviarán los datos a la herramienta).

Por el socket de entrada se irán enviando, uno a uno, los datos del fichero elegido a la herramienta. Por el socket de salida se irán leyendo las respuestas de la herramienta. El hilo correspondiente almacenará en la base de datos cada par enviado/recibido (para el usuario actual) y enviará al cliente dicho par.

En el lado del cliente, al recibir el mensaje del servidor, se dibujará en la gráfica el par recibido.

➤ **endBooking**

Esta petición se envía al servidor si el usuario decide salir de la reserva o ésta termina por sí sola (el tiempo se ha agotado) mientras el usuario se encuentra en la página de ejecución. Se realiza antes de preguntar si se desean guardar los datos. Su finalidad es detener las herramientas que el usuario esté usando y, en caso de que el tiempo se haya agotado, borrar la reserva de la base de datos.

El cliente envía al servidor un objeto JSON con los campos *operation*, *motive*, *token*, *password* y *user*. Estos campos tendrán los valores:

- *operation*: "endBooking"
- *user*: el usuario guardado en *sessionStorage* tras el inicio de sesión
- *password*: la contraseña guardada en *sessionStorage* tras el inicio de sesión
- *token*: token asignado al usuario tras el inicio de sesión

- *motive*: motivo de la parada (el usuario o la finalización normal de la reserva)

El servidor recibe la petición del usuario y, para la operación *endBooking*, el servidor cierra los sockets que está utilizando el usuario, se notifica a los laboratorios y se elimina el token. Luego se crea un nuevo token para nuevas ejecuciones. En el caso de que el motivo de la parada sea la finalización de la reserva, se hace una consulta a la base de datos eliminando de la tabla booking la reserva. El servidor envía al usuario el nuevo token.

En el lado del cliente, se almacena el nuevo token y se procede a ofrecer la posibilidad de guardar los datos al usuario.

5.2. Herramientas

En este proyecto se utilizan dos herramientas: *RLF_Cam* y *RLF_Chart*. Estas herramientas se han creado según las especificaciones del proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha. A continuación se describirá el diseño y funcionalidad de las herramientas mencionadas.

- **RLF_Cam**: su función será activar la cámara que grabará en video al motor, video que será transmitido en la página web de ejecución de la reserva. Tendrá tres constantes: la IP (ip) y el puerto (port) a dónde se enviará el streaming de la cámara, y la duración del video (*video_length*) que será el número de segundos correspondientes a la duración de la reserva. La única acción posible sobre la herramienta será la de comenzar a transmitir (*stream*).

La herramienta, primeramente, recuperará las constantes que fueron definidas en su creación. Utilizando el puerto e IP recuperados, la duración del video y el VLC Player, se activará con este programa la cámara de la máquina donde se encuentra la herramienta. Se transmitirá este video, mediante el protocolo *rtp*, al IP y puerto indicados, durante el tiempo de reserva definido. En un futuro, lo que grabará la cámara será el motor, pero dado que no se cuenta con éste, se

grabarán los valores que se muestran en la ventana que, solamente para esta simulación, lanzará la herramienta *RLF_Chart*.

- **RLF_Chart:** su función será recibir la configuración elegida por el usuario para el PID y, para cada valor enviado posteriormente por el usuario, enviar una respuesta al servidor.

Tendrá tres parámetros de entrada: los parámetros Proporcional, Integral y Derivativo enviados por el usuario, y una constante: la cantidad de datos que recibirá. Los datos enviados/recibidos tras esto, serán fruto de una comunicación directa con el servidor de aplicaciones, sin pasar por el laboratorio.

La herramienta, primeramente, recuperará los parámetros y la constante definidos en su creación.

En un futuro, la herramienta, para cada dato que vaya recibiendo y utilizando la configuración dada, calculará la salida correspondiente a través de un PID, se comunicará con el motor y devolverá las respuestas al servidor. Dado que aún no está implementada la comunicación herramienta-motor, se realizará una simulación, como modo de comprobar tanto el funcionamiento de *RLF_Chart* como de *RLF_Cam*.

La herramienta *RLF_Chart*, para cada valor enviado del fichero de datos elegido por el usuario, devolverá al servidor el valor enviado 5 posiciones antes. Por otro lado, solamente para probar el funcionamiento de *RLF_Cam*, lanzará una ventana en la que se mostrará cada valor recibido y su correspondiente respuesta (*RLF_Cam* activará la cámara que grabará esta ventana).

5.3. Diagrama de despliegue

El siguiente diagrama muestra el despliegue en distintos dispositivos de los componentes identificados en la sección 4.1.

Los usuarios deberán conectarse a la aplicación desde ordenadores cuyos navegadores tengan instalado Java. El servidor que atenderá las peticiones de los usuarios será un

servidor de aplicaciones Glassfish y la comunicación cliente-servidor se realizará a través de WebSockets. El servidor contendrá las páginas web de la aplicación, creadas con HTML5 y realizará consultas a una base de datos MySQL. Dicho servidor podrá, así mismo, conectarse a laboratorios. Los laboratorios serán ejecutados a través de una aplicación Java, usarán la misma base de datos MySQL que el servidor y ejecutarán, a petición del servidor, herramientas que, a su vez, gestionarán sus parámetros de salida y entrada a través de una base de datos propia SQLite.

Es necesario aclarar que el funcionamiento de los laboratorios y herramientas existía previamente a este proyecto. La aplicación Java que gestiona los laboratorios y herramientas, y la librería con los métodos para la lectura/guardado de parámetros de las herramientas, son parte del proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha.

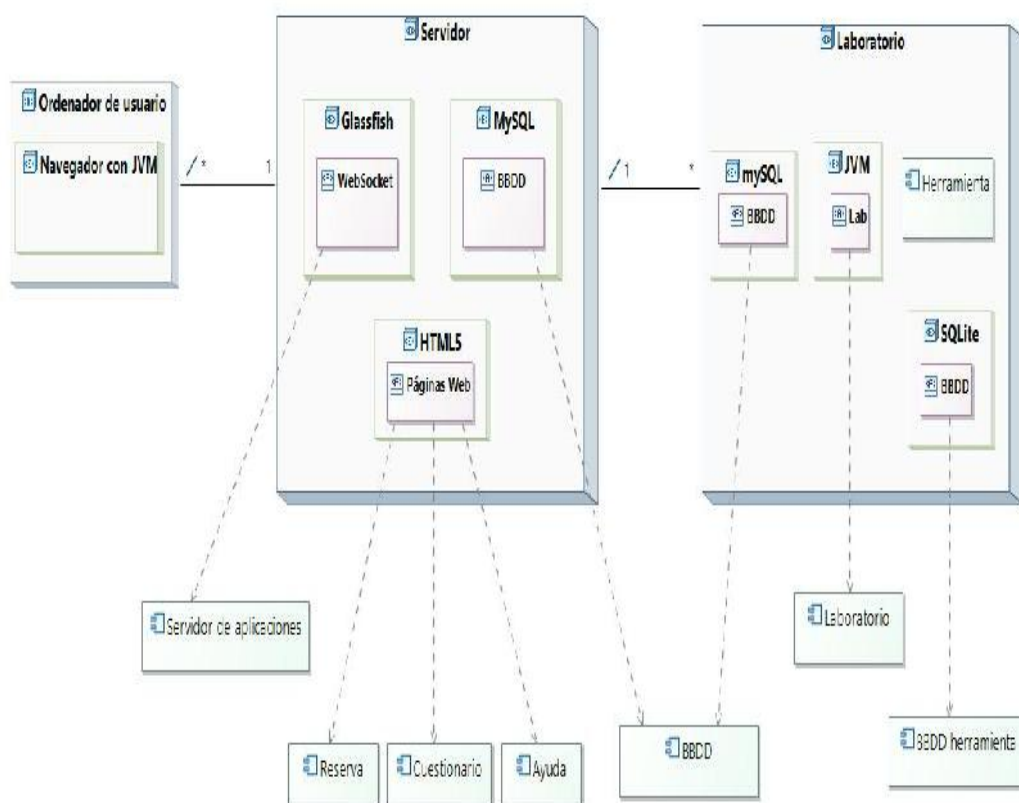


Ilustración 52: Diagrama de despliegue [27]

6. Pruebas

En este apartado se especificarán las pruebas que se han realizado sobre el sistema para comprobar su correcto funcionamiento.

6.1. Pruebas de funcionalidad

Estas pruebas verifican que la funcionalidad del sistema coincida con los requisitos anteriormente definidos en la fase de análisis. Dado que algunas de estas pruebas generaron errores inicialmente, estos errores fueron corregidos y las pruebas fueron repetidas hasta alcanzar los resultados previstos.

Las pruebas de funcionalidad serán recogidas en tablas con el siguiente formato:

ID	PF-XX
Nombre	
Resultado	
Descripción	
Acciones	

Tabla 118: Prueba de funcionalidad

- **ID:** identificador único de la prueba. Su nomenclatura es la siguiente:
 - **PF:** siglas correspondientes a “Prueba de Funcionalidad”.
 - **XX:** numeración consecutiva de las pruebas, comenzando por 01.
- **Nombre:** nombre de la prueba
- **Resultado:** indicación de si el resultado de la prueba es el previsto. Puede tomar los valores “Válido” o “No válido”.
- **Descripción:** descripción de la finalidad de la prueba.
- **Acciones:** pasos para la ejecución de la prueba.

ID	PF-01
Nombre	Prueba de funcionalidad 1
Resultado	Válido
Descripción	Comprobación de los servicios de inicio de sesión, cierre de sesión y visualización de reservas.
Acciones	<ol style="list-style-type: none"> 1. Acceder a la aplicación desde un navegador web. 2. Comprobar que no se brinda la opción de registrarse en la pantalla de inicio de sesión. 3. Introducir y enviar un usuario y/o contraseña que no esté registrado en la BBDD. 4. Comprobar que la aplicación muestra un mensaje de error. 5. Introducir y enviar un usuario y/o contraseña registrado en la BBDD. 6. Comprobar que la aplicación inicia la sesión del usuario y envía a la página con el listado de reservas. 7. Comprobar que la aplicación brinda la opción de cerrar la sesión. 8. Hacer los pasos anteriores para otros sistemas operativos y comprobar que el resultado es el mismo.

Tabla 119: Prueba PF-01

ID	PF-02
Nombre	Prueba de funcionalidad 2
Resultado	Válido
Descripción	Comprobación de que el menú de opciones se encuentra en cada página.
Acciones	<ol style="list-style-type: none"> 1. Una vez en la página del listado de reservas, comprobar que el menú de opciones está en esta página. 2. Seleccionar la opción Reservar del menú y comprobar que el menú se encuentra. 3. Seleccionar la opción Ayuda del menú y comprobar que el menú se encuentra. 4. Seleccionar la opción Cuestionario del menú y comprobar que el menú se encuentra.

Tabla 120: Prueba PF-02

ID	PF-03
Nombre	Prueba de funcionalidad 3
Resultado	Válido
Descripción	Comprobación de que se cargan los temas de ayuda en la página de ayuda.
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción Ayuda del menú y comprobar que se muestran varios títulos de temas de ayuda. 2. Seleccionar distintos títulos y ver que se muestra contenido de ayuda relacionado.

Tabla 121: Prueba PF-03

ID	PF-04
Nombre	Prueba de funcionalidad 4
Resultado	Válido
Descripción	Comprobación de que se carga el cuestionario en la página correspondiente y se puede enviar éste.
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción Cuestionario del menú y comprobar que se muestran varias preguntas con un listado de respuestas para cada una. 2. Seleccionar cualquier otra opción del menú y comprobar que se puede salir del cuestionario sin enviarlo. 3. Seleccionar la opción Cuestionario nuevamente. 4. Comprobar que cada pregunta tiene una respuesta marcada por defecto. 5. Pulsar el botón Enviar y comprobar que la aplicación muestra un mensaje de confirmación del envío. 6. Repetir varias veces el envío y comprobar que no se producen errores.

Tabla 122: Prueba PF-04

ID	PF-05
Nombre	Prueba de funcionalidad 5
Resultado	Válido
Descripción	Comprobación de las funcionalidades y restricciones del calendario de reservas.
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción Reserva del menú y comprobar que se muestran un calendario de lunes a viernes, con horario de 9 a 21. 2. Utilizar las flechas del calendario y comprobar que se puede cambiar la semana. 3. Seleccionar una franja en rojo y comprobar que la aplicación no permite reservar esa franja. 4. Seleccionar una franja con fecha pasada y comprobar que la aplicación muestra un mensaje de error por fecha antigua. 5. Seleccionar una franja no ocupada por ninguna reserva y comprobar que la aplicación muestra un mensaje de confirmación y la franja queda marcada en verde en el calendario. 6. Seleccionar una franja no ocupada por ninguna reserva y comprobar que la aplicación cambia la fecha de la reserva previa. 7. Salir del calendario, esperar a que la reserva comience y volver al calendario. Comprobar que la reserva comenzada aparece en gris. 8. Seleccionar cualquier franja libre y comprobar que la aplicación muestra un mensaje de error

por reserva ya comenzada.

Tabla 123: Prueba PF-05

ID	PF-06
Nombre	Prueba de funcionalidad 6
Resultado	Válido
Descripción	Comprobación de las cuentas regresivas y botones del listado de reservas.
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción Mis reservas del menú y comprobar que se muestra un listado con las reservas realizadas por el usuario o una lista vacía en caso de no tener reservas. 2. Si hay reservas, comprobar que el tiempo restante para inicio que la cuenta regresiva muestra es correcto. 3. Pulsar el botón Eliminar, confirmar la eliminación y comprobar que la reserva se ha borrado de la lista. 4. Hacer otra reserva y comprobar que se carga en el listado. 5. Esperar a que la cuenta regresiva del tiempo restante para inicio llegue a 0. 6. Comprobar que el botón Eliminar es sustituido por el botón Comenzar y que ya no se permite la eliminación de la reserva. 7. Comprobar que se muestra una cuenta regresiva con el tiempo de reserva restante.

Tabla 124: Prueba PF-06

ID	PF-07
Nombre	Prueba de funcionalidad 7
Resultado	Válido
Descripción	Comprobación de la posibilidad de elección de parámetros para enviar al PID.
Acciones	<ol style="list-style-type: none"> 1. Pulsar el botón Comenzar de la reserva y comprobar que se muestra una página donde se pueden introducir los valores Proporcional, Integral y Derivativo, y se puede escoger un fichero para enviar. 2. No rellenar alguno de los campos, dar a Enviar y comprobar que la aplicación envía un mensaje de error. 3. Elegir un fichero de texto cuyo contenido no sea una lista de valores numéricos y comprobar que la aplicación envía un mensaje de error. 4. Introducir valores válidos en Proporcional, Integral y Derivativo y escoger un fichero con contenido válido. 5. Pulsar Enviar y comprobar que no se producen errores, que se pasa a la página de espera y luego a la de ejecución de la reserva.

Tabla 125: Prueba PF-07

ID	PF-08
Nombre	Prueba de funcionalidad 8
Resultado	Válido
Descripción	Comprobación de la carga de herramientas en la página de ejecución.
Acciones	<ol style="list-style-type: none"> 1- Pulsar Enviar en la página de configuración de parámetros y esperar a que se cargue la página de ejecución de la reserva. 2- Comprobar que se muestra el video del motor, una gráfica vacía, una cuenta regresiva indicando el tiempo restante, botones Comenzar y Salir, y que no se ofrece el menú de opciones. 3- Pulsar Comenzar y comprobar que se comienzan a dibujar dos líneas, punto por punto, en la gráfica.

Tabla 126: Prueba PF-08

ID	PF-09
Nombre	Prueba de funcionalidad 9
Resultado	Válido
Descripción	Comprobación de la parada de herramientas y guardado de datos.
Acciones	<ol style="list-style-type: none"> 1- Pulsar el botón Salir dentro de la página de ejecución de la reserva. 2- Comprobar que la aplicación muestra un mensaje preguntando si se desea guardar los resultados de la ejecución. 3- Elegir Aceptar y comprobar que la aplicación crea un fichero que se puede abrir o guardar en una ruta elegida por el usuario, y que se envía al usuario a la página con el listado de reservas. 4- Ir a la ruta elegida y comprobar que se ha guardado un fichero con los resultados de la ejecución. 5- Entrar a la reserva y pulsar Salir nuevamente. 6- Pulsar Cancelar en el mensaje de guardado y comprobar que la aplicación envía a la página con el listado de reservas.

Tabla 127: Prueba PF-09

ID	PF-10
Nombre	Prueba de funcionalidad 10
Resultado	Válido
Descripción	Comprobación de la finalización de reserva
Acciones	<ol style="list-style-type: none"> 1- Esperar, dentro de la página de ejecución, a que la cuenta regresiva del tiempo de reserva restante llegue a 0. 2- Comprobar que la aplicación muestra un mensaje preguntando si se desea guardar los resultados de la ejecución. 3- Elegir Aceptar y comprobar que la aplicación crea un fichero que se puede abrir o guardar en una ruta elegida por el usuario, y que se envía al usuario a la página con el listado de reservas. 4- Comprobar que la reserva ya no se encuentra en el listado de reservas. 5- Ir a la ruta elegida y comprobar que se ha guardado un fichero con los resultados de la ejecución.

Tabla 128: Prueba PF-10

6.2. Matriz de trazabilidad

Con el objetivo de comprobar que cada requisito de software definido en la fase de análisis ha sido integrado en el sistema, se ha construido la siguiente matriz de trazabilidad. Esta matriz relaciona cada requisito software con la prueba de funcionalidad que demuestra su cumplimiento.

	PF-01	PF-02	PF-03	PF-04	PF-05	PF-06	PF-07	PF-08	PF-09	PF-10
RS-F-01	X									
RS-F-02	X									
RS-F-03	X									
RS-F-04						X				
RS-F-05						X				
RS-F-06						X				
RS-F-07						X				
RS-F-08						X				
RS-F-09						X				
RS-F-10		X								
RS-F-11	X									
RS-F-12			X							
RS-F-13			X							
RS-F-14				X						
RS-F-15				X						
RS-F-16				X						
RS-F-17				X						
RS-F-18				X						
RS-F-19				X						
RS-F-20					X					
RS-F-21					X					
RS-F-22					X					
RS-F-23					X					
RS-F-24					X					
RS-F-25					X					
RS-F-26					X					
RS-F-27					X					
RS-F-28					X					
RS-F-29							X			
RS-F-30							X			
RS-F-31							X			
RS-F-32							X			
RS-F-33								X		
RS-F-34								X		
RS-F-35								X		
RS-F-36								X		
RS-F-37								X		
RS-F-38									X	
RS-F-39								X		
RS-F-40									X	X
RS-F-41										X
RS-F-42									X	X
RS-N-01	X									
RS-N-02	X									
RS-N-03	X									
RS-N-04	X									
RS-N-05	X									
RS-N-06						X				
RS-N-07					X					
RS-N-08						X				
RS-N-09										X

Tabla 129: Matriz de trazabilidad RS-PF

Del análisis de la matriz de trazabilidad se puede concluir, dado que cada RS tiene asociado al menos una PF, que todos los requisitos de software han sido integrados en el sistema.

7. Planificación

En este capítulo se especificará la planificación seguida para el desarrollo de este proyecto. Se mostrarán, para cada fase en la que se ha dividido el proyecto, los días y horas estimadas para su realización.

7.1. Planificación de actividades

La siguiente tabla muestra la lista de actividades realizadas, las fechas de inicio y fin de éstas, y la duración estimada, tanto en días como en horas, de cada una de las tareas. La cantidad de horas trabajadas por día se corresponden con una jornada de trabajo típica y se excluyen los fines de semana.

Actividad	Fecha de inicio	Fecha de fin	Duración (días)	Horas/Día	Duración (horas)
Propuesta	30/01/2013	30/01/2013	1	8	8
Análisis	31/01/2013	04/03/2013	23	8	184
Diseño	05/03/2013	02/04/2013	21	8	168
Implementación	03/04/2013	20/11/2013	166	8	1328
Pruebas	21/11/2013	04/12/2013	10	8	80
Documentación	05/12/2013	10/02/2014	48	8	384

Tabla 130: Planificación de actividades

7.2. Diagrama de Gantt

El siguiente diagrama de Gantt representa la planificación detallada anteriormente:

DESARROLLO DE UNA APLICACIÓN SOFTWARE PARA LABORATORIOS REMOTOS

Ilustración 53: Diagrama de Gantt.

8. Presupuesto

En este capítulo se detalla el presupuesto aproximado necesario para el despliegue de este proyecto. Dicho presupuesto será desglosado en costes orientados a distintos factores influyentes en la realización del proyecto.

8.1. Costes de desarrollo

Los costes de desarrollo no influyen en el despliegue de la aplicación, pero se detallan como indicación de la cuantía necesaria para su creación.

8.1.1. Costes de personal

La siguiente tabla muestra el coste del personal involucrado en la realización del proyecto:

Concepto	Coste/hora (€)	Horas	Coste (€)
Analista	45	576	25920
Diseñador	30	168	5040
Programador	30	1408	42240
TOTAL	105	2152	73200

Tabla 131: Costes de personal

- **Analista:** 72 días*8 horas=576 horas
- **Diseñador:** 21 días*8 horas=168 horas
- **Programador:** 176 días*8 horas=1408 horas

8.1.2. Costes materiales

La siguiente tabla muestra los recursos materiales utilizados en el proyecto:

Concepto	Coste (€)	Meses de dedicación	Período de depreciación (meses)	Coste imputable
Portátil Lenovo G500 i7-3632QM, 2.2 Ghz, 8GB RAM, 500GB HD	569	13	60	123,28
Philips Webcam SPC620NC	20	13	60	4,33
TOTAL	589	13	60	127,61

Tabla 132: Costes materiales

El porcentaje de dedicación al proyecto de los recursos ha sido del 100%. La amortización se ha calculado mediante la fórmula:

Coste imputable = (Meses de dedicación/ Período de depreciación)*Coste * Porcentaje de dedicación (1)

8.1.3. Coste total

Concepto	Coste (€)
Costes de personal	73200
Costes materiales	127,61
Total sin IVA	73327,61
Total con IVA (21%)	88726,408

Tabla 133: Coste total de desarrollo

8.2. Costes de despliegue

Los costes de despliegue describen el gasto necesario para la instalación de la aplicación. Muchos de estos costes han sido sacados del proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha, base de este nuevo proyecto.

8.2.1. Costes del servidor

Concepto	Coste (€)
PowerEdge T410 Tower Server, Dell	1217
SAI	80
TOTAL	1297

Tabla 134: Costes del servidor

8.2.2. Costes del laboratorio

El laboratorio en el que se encontrarán las herramientas utilizadas en este proyecto ocasionará el siguiente gasto [1]:

Concepto	Coste (€)
PowerEdge T110, Dell	378
TOTAL	378

Tabla 135: Costes del laboratorio

8.2.3. Costes de las herramientas

Para las nuevas herramientas que se han creado en este proyecto, los costes son los siguientes [1]:

Concepto	Coste (€)
eLight HD 720p Webcam, Trust	80
TOTAL	80

Tabla 136: Costes de las herramientas

8.2.4. Coste total

Concepto	Coste (€)
Costes del servidor	1297
Costes del laboratorio	378
Costes de las herramientas	80
Total sin IVA	1755
Total con IVA (21%)	2123,55

Tabla 137: Coste total de despliegue

9. Conclusiones

Una vez desarrollado el proyecto, en este capítulo se analizarán los logros con respecto a los objetivos definidos inicialmente y los trabajos que se plantean en un futuro para la optimización y crecimiento de la aplicación desarrollada. Se explicarán, además, los problemas surgidos durante la implementación y las conclusiones personales del desarrollador.

9.1. Conclusiones generales

En este proyecto se ha desarrollado una aplicación web intuitiva y de fácil uso, que permitirá realizar prácticas, utilizando recursos ubicados remotamente. A pesar de que la aplicación aborda una práctica específica, se ha implementado de un modo que permita la gestión de más prácticas en un futuro, llevando a cabo modificaciones mínimas en el código.

Se ha logrado crear un sistema que permite al usuario elegir la fecha en la que desea realizar la práctica y garantizar la exclusividad de uso de dicho usuario sobre las herramientas que componen la práctica.

Se han creado dos herramientas específicas, siguiendo la guía proporcionada en el proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha. A través de estas herramientas, se ha logrado crear una interfaz que proporciona al usuario el video en tiempo real de un motor ubicado remotamente, y una gráfica que dibuja uno a uno los datos que se envían y reciben de una herramienta ubicada en un laboratorio remoto. Estos datos servirán para un posterior control del motor, que será implementado en proyectos futuros.

La aplicación permite al usuario rellenar y enviar un cuestionario sobre la práctica realizada, cuyas respuestas serán utilizadas con fines estadísticos. Adicionalmente, se ofrecen temas de ayuda al usuario sobre el uso de la aplicación, para un mejor aprovechamiento de sus servicios.

El uso de HTML5 para el desarrollo de las páginas permitió, además de contar con el último estándar HTML, disponer del servicio Web Storage, que permitió guardar mucha información de sesión en el navegador sin afectar el rendimiento de la aplicación. La utilización de WebSockets permitió la comunicación bidireccional entre cliente y servidor, de modo que la cuestión de que el servidor pudiese enviar datos al cliente sin haber mediado una petición previa, fue solucionada inmediatamente a través de la utilización de esta tecnología. El uso de jQuery contribuyó enormemente tanto al ahorro de tiempo en la creación de las interfaces de usuario, como al aspecto visual de la aplicación.

Durante el desarrollo de la aplicación han surgido muchos problemas que, en su mayoría, han encontrado una solución rápidamente. Sin embargo, las mayores dificultades surgieron

al inicio del todo y el final de la implementación. Este proyecto tenía otro como base, en el que se plantearon los conceptos “laboratorio” y “herramienta”. La aplicación web tenía que comunicarse con un laboratorio, pero no se podía modificar nada de la implementación de éste, sino que tenía que adaptarse a estructuras ya creadas. Esto supuso, a pesar de contar con la documentación y el código fuente del proyecto anterior, muchas horas de investigación, lectura, navegación por el código y pruebas de funcionamiento. Adicionalmente, la aplicación no podía simplemente activar una cámara o enviar unos datos, sino que todo tenía que ser ejecutado por herramientas, estructuras que debían seguir un formato específico, lanzadas por los laboratorios.

Por otro lado, al final de la implementación, con las comunicaciones servidor-laboratorio y servidor-herramienta funcionando, así como el resto de las funcionalidades, aún no se lograba ver el video en la página correspondiente. La búsqueda de una solución llevó muchísimas horas, primero porque el VLC daba un error continuamente (que se solucionó desinstalando paquetes que en la fase de investigación se habían instalado, pero ya no eran necesarios, e instalando otros) y luego porque el video enviado se transmitía desde que se había activado la cámara en la página y no desde el momento actual (solucionado al cambiar de protocolo para el video: de HTTP a RTP).

A pesar de estos problemas, finalmente se logró la implementación de la aplicación, abarcando todos los objetivos propuestos.

9.2. Conclusiones personales

Este proyecto ha sido la prueba final de 5 años de carrera. En los 4 años anteriores me tuve que enfrentar a prácticas que, o bien eran pequeñas y simples, o se realizaban en grupo y con una intensa supervisión de los profesores, o simplemente todo el proceso estaba explicado y solo había que seguir el camino indicado. Este proyecto, sin embargo, constituye mi primer enfrentamiento real con la toma de decisiones, con la investigación exhaustiva sobre tecnologías a utilizar y con la documentación completa y detallada de un proyecto.

Durante la carrera se me han ofrecido pedazos de información. Por un lado estaban las interfaces de usuario, por otro las bases de datos, por otro los sistemas distribuidos, las asignaturas sobre documentación y así muchos más trozos que me preguntaba cómo

encajarían. Este proyecto me ha permitido auto responderme a esa pregunta, al haber tenido que utilizar todos estos recursos que fui aprendiendo en la carrera, para desarrollar una aplicación que finalmente los abarcase todos.

Responder a esa pregunta era fundamental para consolidar los conocimientos adquiridos en estos 5 años. Gracias a este proyecto, eso ha sido posible.

9.3. Trabajos futuros

Este proyecto es una pequeña parte de una idea mayor y más compleja. Las bases fueron definidas en el proyecto “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos” de Carlos A. Rodríguez Mecha, en el cual se implementaba la aplicación para gestionar un laboratorio remoto y se definía el procedimiento a seguir para la creación de herramientas. En el proyecto actual se ha incluido una aplicación web que permite la realización de una práctica específica y para ello se han creado dos herramientas que se podrán reutilizar en otras prácticas, si fuesen necesarias.

Debido a la complejidad de la idea general y ausencia del tiempo necesario para la implementación, ha quedado pendiente la comunicación entre la herramienta (software) que recoge los datos enviados al motor (hardware) y éste. Esta misma complejidad da cabida a la posibilidad de incorporar otras funcionalidades, como las siguientes:

- **Soporte para varios motores:** una vez que se haya implementado la comunicación herramienta-motor, puede ocurrir que se tenga más de un motor (por lo tanto, más de un laboratorio con las mismas herramientas) que ofrecer a los estudiantes para hacer la práctica. En este caso, se debería permitir que usuarios distintos reservasen una misma franja de tiempo, siempre y cuando los laboratorios reservados fuesen diferentes (el laboratorio debería ser asignado por la aplicación, no elegido por el usuario, dado que los servicios serían iguales para cada laboratorio).
- **Gestión de varias prácticas:** la aplicación permitiría realizar varias prácticas distintas y, para ello, mostraría un listado de las prácticas que gestiona. El usuario, en dependencia de la práctica deseada, elegiría una para reservarla y se le mostraría un calendario específico de la práctica elegida. Consecuentemente, se permitiría la multiplicidad de reservas, pero de prácticas distintas. Por otro lado, cada práctica

tendría sus propias herramientas cuyas funcionalidades dependerían del objetivo de la práctica.

- **Evaluación de la práctica:** se incorporarían una serie de preguntas sobre la práctica para que, tras la realización de ésta, el usuario las respondiese, enviase y fuesen posteriormente evaluadas por un profesor. En el caso de ser un cuestionario, la propia aplicación podría consultar la respuesta correcta para cada pregunta, calcular la nota e informar al usuario de los resultados de su evaluación.
- **Historial de resultados:** la aplicación ofrecería al usuario un historial con los resultados obtenidos tras la realización de cada práctica. De este modo se garantizaría que el usuario contase con estos datos, en caso que hubiese decidido no guardarlos en un fichero de texto tras finalizar la reserva.
- **Recordatorio de reserva:** se enviaría un correo electrónico al usuario avisándole de la proximidad o inicio de su reserva.
- **Registro de usuarios:** se permitiría el registro de usuarios en la aplicación, en la página de inicio de sesión. Esta funcionalidad sería necesaria si se deseara usar la aplicación para un entorno externo a la universidad, en el que los usuarios no estuviesen en una base de datos.

Anexo I: Guía de usuario

Este anexo está concebido como una guía de navegación por la aplicación para el usuario de ésta. Se explicará cómo utilizar cada servicio de la aplicación y los errores que pueden ocurrir durante la ejecución.

Inicio de sesión

La primera página que ve el usuario es la que permite iniciar sesión. El usuario debe introducir un nombre de usuario y contraseña válidos en los campos Usuario y Contraseña que brinda la aplicación, y pulsar el botón Entrar. Si la aplicación muestra un mensaje indicando que el nombre de usuario y/o la contraseña son incorrectos, el usuario debe intentar nuevamente iniciar la sesión, corrigiendo los campos. Si el usuario y la contraseña introducidos son correctos, la aplicación inicia la sesión del usuario y éste es dirigido a la página con el listado de sus reservas (lista vacía, si el usuario no ha hecho ninguna reserva).

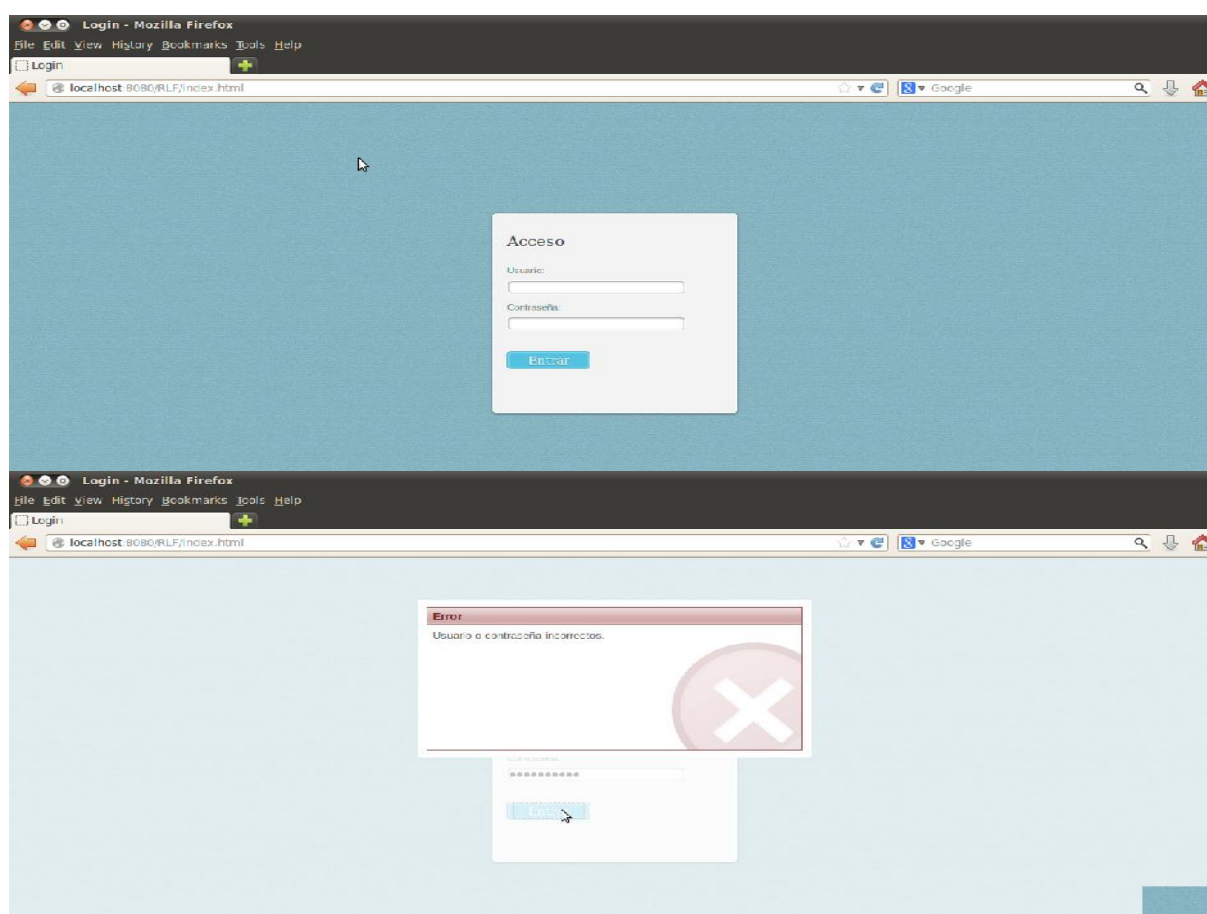


Ilustración 54: Guía de usuario - Inicio de sesión

Cierre de sesión

En cada página de la aplicación existe un ícono de OFF en la parte superior derecha. El usuario puede cerrar su sesión en cualquier momento, pulsando ese ícono. El usuario, para salir de la aplicación, debe siempre usar esta opción, de lo contrario su sesión puede quedar

bloqueada.

Acceso a temas de ayuda

En cada página de la aplicación (excepto en la de ejecución de la reserva), el usuario puede acceder a los temas de ayuda que se brindan, pulsando la opción Ayuda del menú que se encuentra en la parte izquierda de la página. Para ver los contenidos de cada tema de ayuda, el usuario debe hacer clic sobre el tema elegido. Si hace clic sobre otro tema, se cierra el tema abierto anteriormente y se abre el seleccionado.

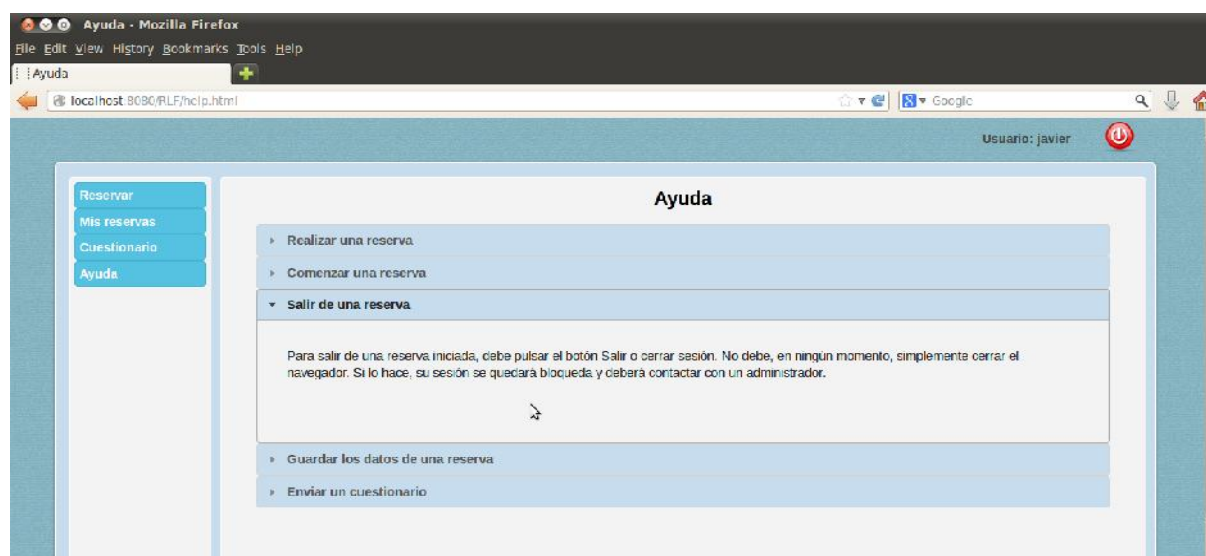


Ilustración 55: Guía de usuario - Ayuda

Envío de cuestionario

En cada página de la aplicación (excepto en la de ejecución de la reserva), el usuario puede acceder al cuestionario sobre la práctica realizada, pulsando la opción Cuestionario del menú que se encuentra en la parte izquierda de la página. Tanto el relleno como el envío del cuestionario son de carácter opcional. El usuario puede salir del cuestionario sin enviarlo. El cuestionario tiene, para cada pregunta, una respuesta marcada por defecto. Para cambiar la respuesta, el usuario debe seleccionar él mismo la respuesta que considere correcta. Cuando haya rellenado el cuestionario en su totalidad, el usuario puede enviar el cuestionario pulsando el botón Enviar. Tras el mensaje de confirmación del envío que se muestra, el usuario puede enviar de nuevo el cuestionario (tantas veces como crea necesario, pero solo es válido el último envío) o ir a cualquier otra opción del menú de la aplicación.

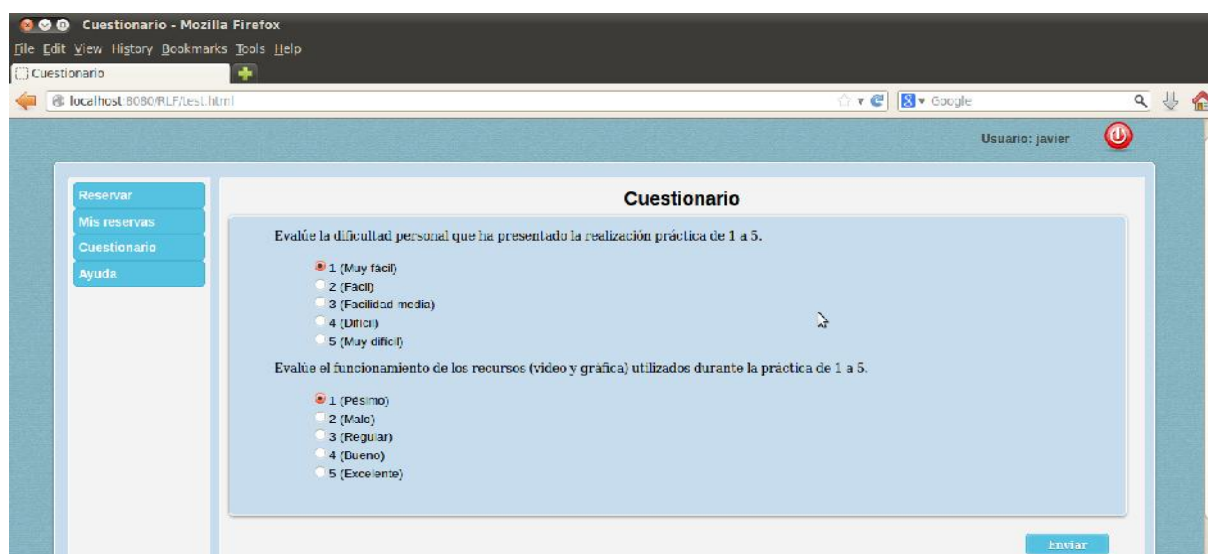


Ilustración 56: Guía de usuario - Cuestionario

Reserva de herramientas

En cada página de la aplicación (excepto en la de ejecución de la reserva), el usuario puede acceder al calendario de reservas, pulsando la opción Reservar del menú que se encuentra en la parte izquierda de la página. El calendario que muestra esta opción solo permite hacer reservas de lunes a viernes, de 9 a 21. Un día con color amarillo, indica el día actual. El usuario puede elegir alguna franja para reservar de la semana que se muestra (la actual), o hacer una reserva para otra semana. Con este fin, en la parte superior izquierda del calendario se muestran dos flechas que permiten pasar a la semana siguiente o posterior. Si el usuario desea regresar directamente a la semana actual, debe pulsar la opción Hoy, situada al lado de las flechas. Esta opción le devolverá a la semana actual. Así, mismo, si en vez de pasar semana por semana, desea ver día a día, debe pulsar la opción Día, situada en la parte superior derecha del calendario, y repetir el mismo procedimiento. Si desea regresar a la vista semanal, debe pulsar Semana.

En cuanto al contenido del calendario, las franjas tienen un rango de 15 minutos (esto significa que la reserva dura 15 minutos). Los colores de las franjas tienen los siguientes significados:

- **Blanco o amarillo (en caso de ser el día actual):** franja libre para reserva.
- **Rojo:** franja reservada por otro usuario.

- **Verde:** franja reservada por el propio usuario.
- **Gris:** franja reservada, pero reserva ya comenzada.

El usuario puede hacer clic sobre cualquier franja, pero existen las siguientes restricciones:

- Si hace clic sobre una franja roja, no ocurre nada, porque dicha franja está ya reservada por otro usuario y no se permiten reservas simultáneas.
- Si hace clic sobre una franja de tiempo que es anterior al momento actual (franja pasada), se muestra un mensaje de error, porque no se permite reservar para una fecha pasada.
- Si, habiendo comenzado una reserva del propio usuario, hace clic sobre cualquier franja, se muestra un mensaje de error, porque debe esperar a que termine su reserva (no se permiten reservas múltiples).

Si ninguno de estos casos se da, se muestra un mensaje de confirmación de la reserva.

El usuario puede decidir cambiar la fecha de la reserva. Para ello, basta con hacer clic en otra franja válida y la reserva se cambiará para esa fecha.

Para comprobar si la reserva ha sido, en efecto, realizada, o simplemente para ver los datos de ésta, el usuario puede ir a la opción Mis Reservas del menú que existe en cada página. Esta opción le muestra la fecha, estado y tiempo restante para el inicio de su reserva. El estado indica si se puede acceder a la reserva ya o no (Lista o En espera, respectivamente). La cuenta regresiva indica el tiempo que queda para que comience la reserva del usuario. Adicionalmente, la reserva muestra un botón Eliminar que el usuario puede pulsar si desea cancelar la reserva completamente. Si este es el caso, debe confirmar esta decisión pulsando Aceptar en el pop-up que se mostrará. Si pulsa Cancelar, no se borra la reserva.

Cuando la cuenta regresiva de tiempo restante para inicio llegue a 0, el estado de la reserva cambia a Lista, aparece una nueva cuenta regresiva, que indica el tiempo de reserva restante (descendiendo desde 15 minutos) y el botón Eliminar es sustituido por un botón Comenzar.

El usuario no puede, a partir de que la reserva comienza, eliminarla, y si desea hacer otra reserva, debe esperar a que la actual termine.

DESARROLLO DE UNA APLICACIÓN SOFTWARE PARA LABORATORIOS REMOTOS

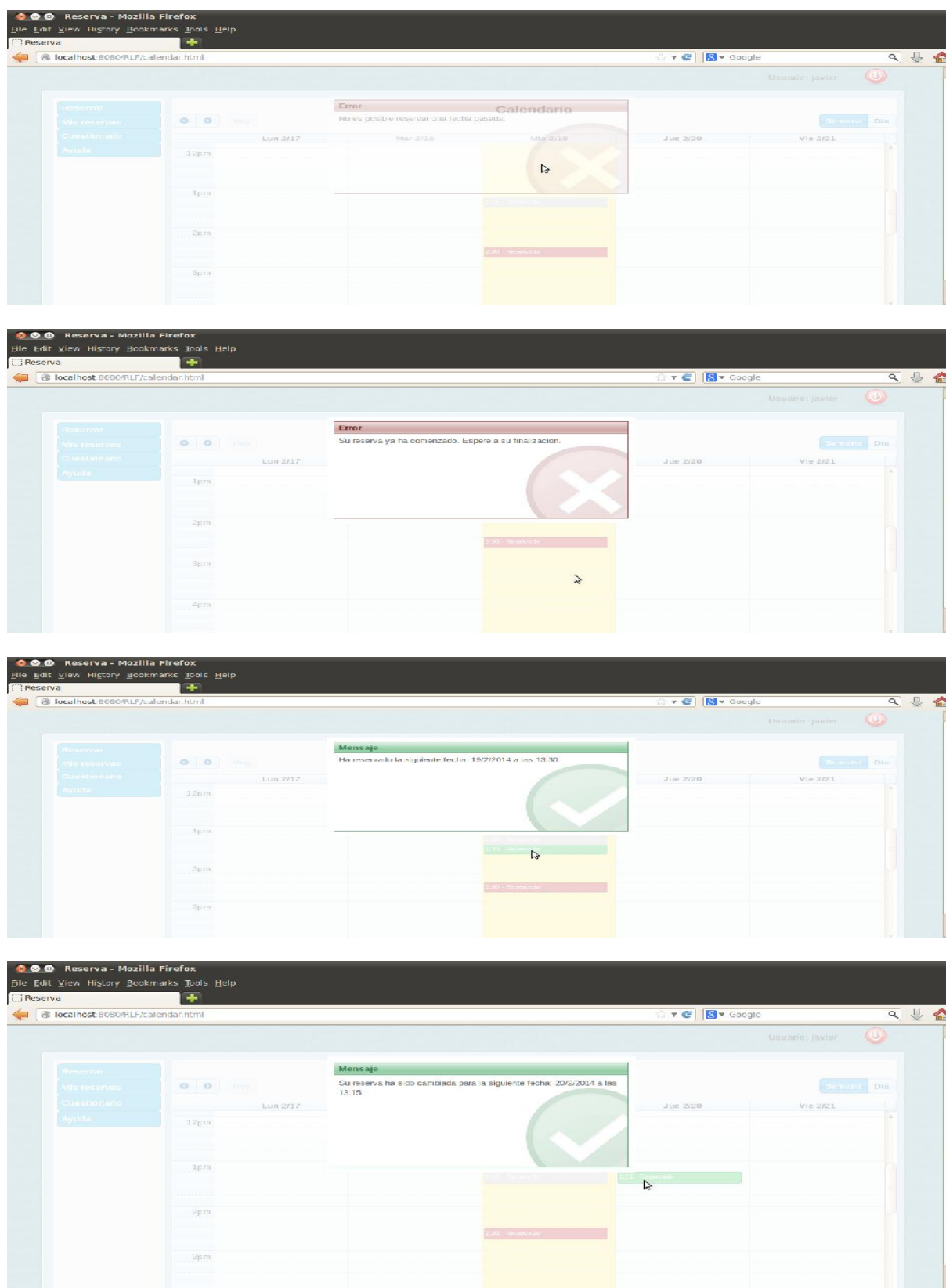


Ilustración 57: Guía de usuario - Reserva de herramientas

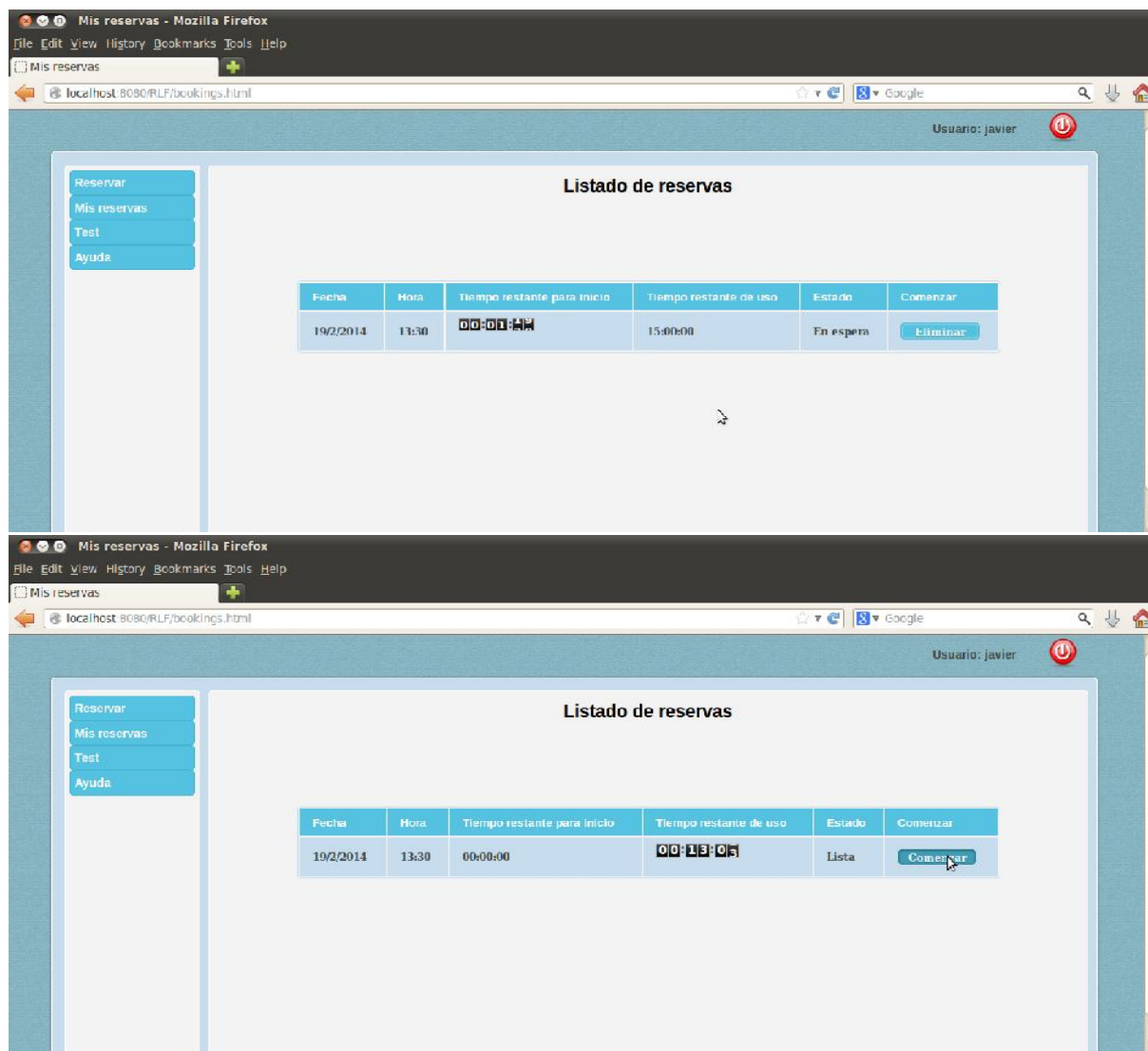


Ilustración 58: Guía de usuario - Estados de reserva

Inicio de reserva

Para iniciar la reserva, el usuario debe pulsar el botón Comenzar de la reserva. Una vez pulsado el botón, se muestra una página donde el usuario debe introducir los valores Proporcional, Integral y Derivativo para enviar al PID. Estos valores (numéricos) deben haber sido calculados anteriormente por el usuario (como parte del trabajo que la práctica conlleva). Así mismo, el usuario debe elegir un fichero de texto elaborado por él mismo para enviar valores al motor. El fichero debe tener un número por línea. Si los valores de Proporcional, Integral y Derivativo, o los datos del fichero, no son numéricos y el usuario pulsa Enviar, se muestra un mensaje de error. Solo se puede acceder a la página de ejecución de la reserva si todos los parámetros son correctos.

Cuando se pulsa Enviar (y los parámetros son correctos), el usuario ve una página de espera antes de que se cargue la página de ejecución. Una vez en esta última, existen varias zonas:

- **Video:** muestra el video en tiempo real del motor. El usuario puede observar las reacciones del motor a la configuración y datos elegidos anteriormente por el usuario.
- **Gráfica:** inicialmente no muestra nada. Cuando se pulse el botón Comenzar y se inicie el envío uno a uno de los datos contenido en el fichero elegido por el usuario, comienzan a dibujarse en la gráfica dos líneas. La línea azul corresponde a los datos enviados y la amarilla a los recibidos como respuesta.
- **Botón Comenzar:** inicia el envío y recepción de datos del motor. Si no se pulsa, la gráfica permanece vacía. Cuando se pulsa desaparece y, en la zona en la que se encontraba, se muestra el dato enviado actual, su respuesta, y así sucesivamente.
- **Cuenta regresiva:** muestra el tiempo que falta para que termine la reserva.
- **Botón Salir:** permite salir de la reserva. A diferencia del resto de páginas, en la que había un menú, en esta página el menú se elimina y en su lugar se muestra el botón Salir. La salida de la reserva se debe realizar a través de este botón, para evitar posibles problemas en la ejecución de las herramientas.

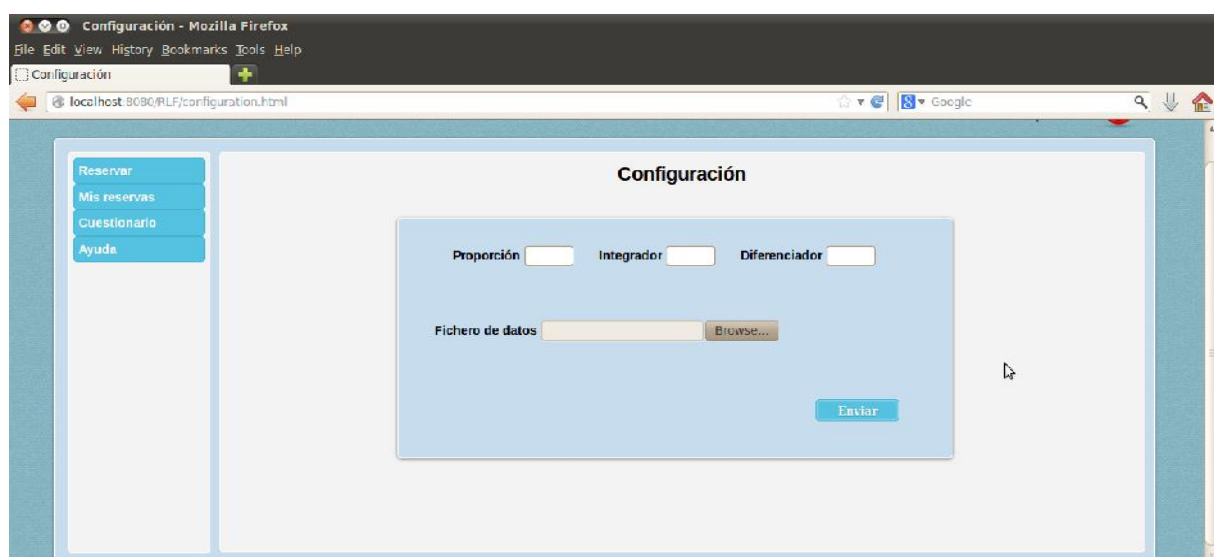


Ilustración 59: Guía de usuario - Configuración

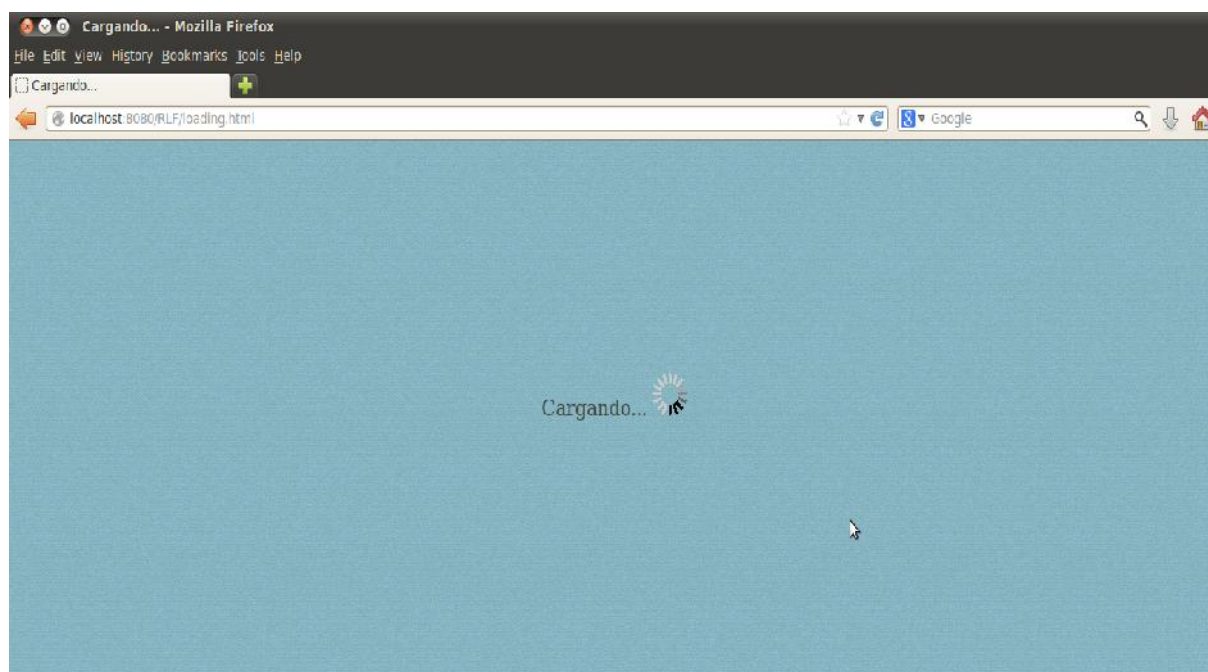


Ilustración 60: Guía de usuario - Página de espera

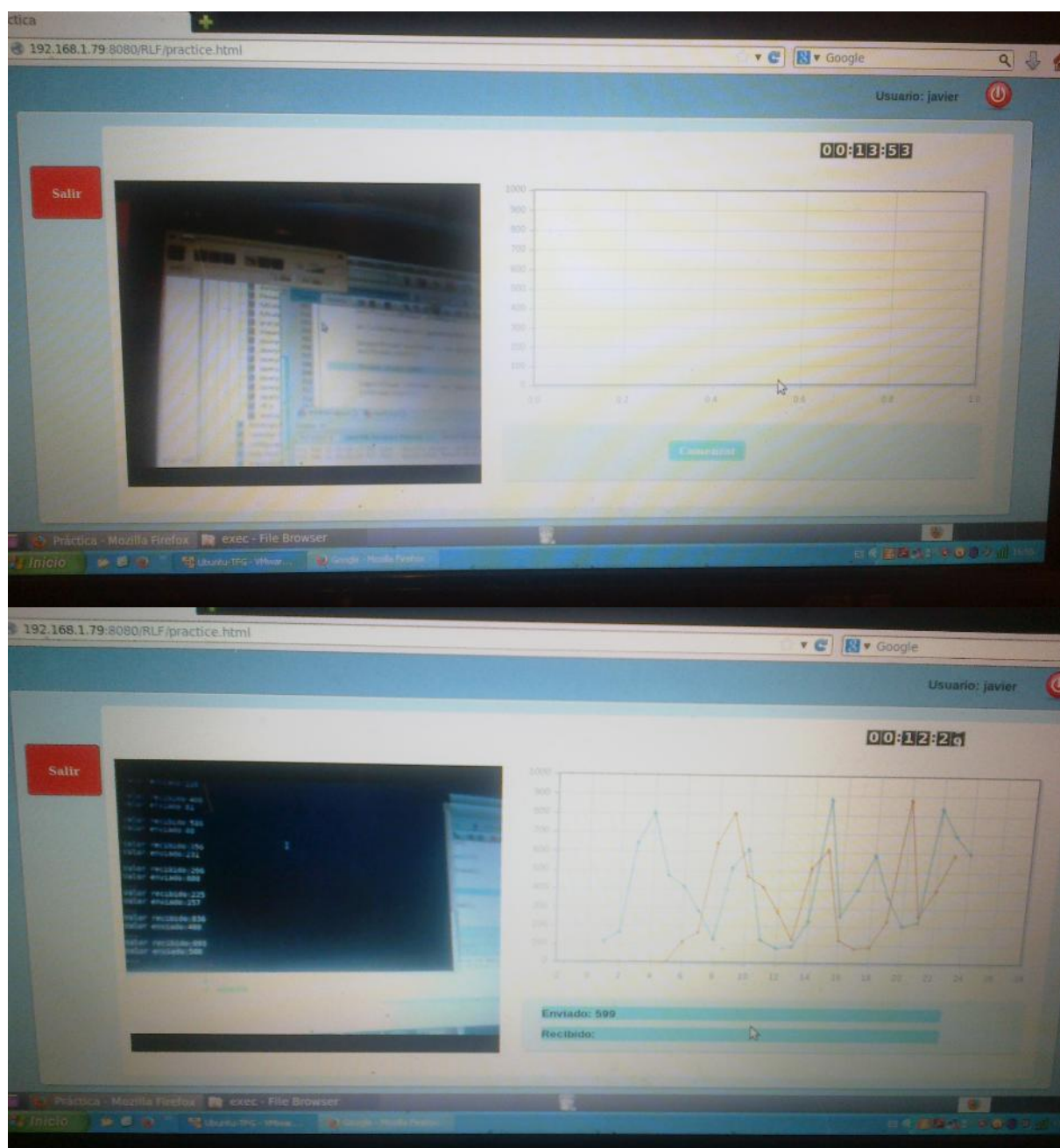


Ilustración 61: Guía de usuario - Ejecución

Salida de reserva y guardado de datos

Al pulsar el botón Salir, además de pararse las herramientas que el usuario está usando, a éste se le ofrece la posibilidad de guardar los datos resultantes de la ejecución (datos enviados y recibidos). Existen las siguientes alternativas:

- Pulsar Aceptar: se crea un fichero de nombre Usuario + Fecha actual, que el usuario puede abrir o guardar en la ruta de su elección. El contenido del fichero serán los datos resultantes de la ejecución de las herramientas.
- Pulsar Cancelar: no se guardan los datos.

En ambos casos, el usuario es enviado como paso final a la página de Mis reservas. El usuario puede entrar de nuevo a la reserva si así lo desea, pulsando el botón Comenzar.

Por otro lado si, estando en la página de ejecución, la cuenta regresiva llega a 0, se ofrecerá también la posibilidad de guardar los datos. Una vez finalizado el tiempo de reserva, para poder acceder a las herramientas nuevamente se debe hacer otra reserva.

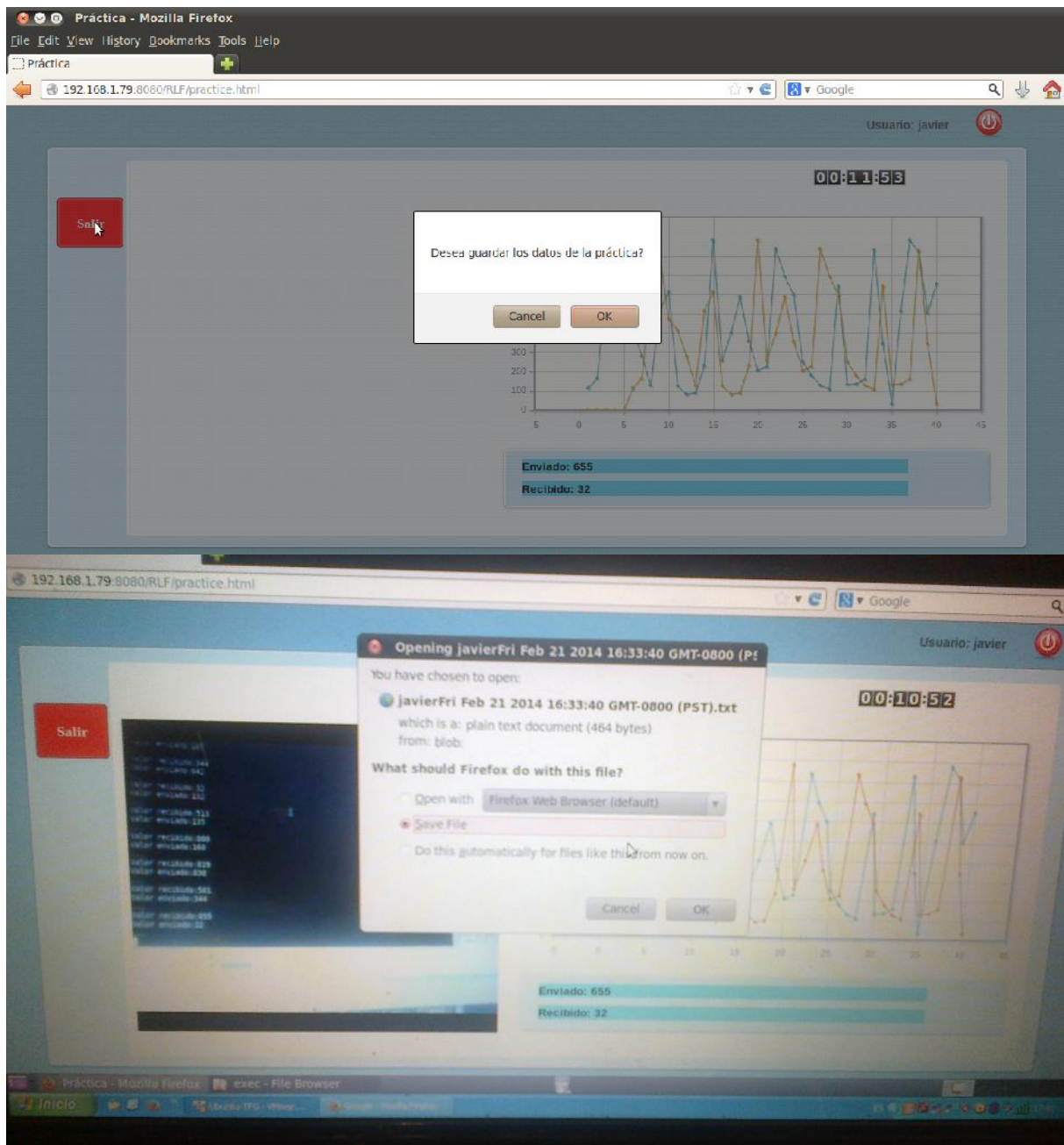


Ilustración 62: Guía de usuario - Guardado de datos

Anexo II: Guía de despliegue

Este anexo indica los pasos a seguir para la instalación y correcta configuración de la aplicación.

Navegador de usuario

La aplicación es multiplataforma y multinavegador. Sin embargo, para el acceso a la aplicación se deben cumplir los siguientes requisitos:

- El ordenador del usuario debe tener acceso a Internet.
- El ordenador del usuario debe tener Java instalado. Se puede descargar desde la página <http://www.java.com/es/>.
- El navegador utilizado debe tener soporte para WebSockets y HTML5. Se puede comprobar si el navegador es compatible con WebSockets en la página <http://www.websocket.org/echo.html>.

Laboratorio

Las instrucciones para el despliegue de un laboratorio se especifican en la memoria del proyecto *“Diseño y desarrollo de un prototipo de Framework para laboratorios remotos”* de Carlos A. Rodríguez Mecha.

Sobre un laboratorio previamente implantado en una distribución Linux, se deberán modificar los siguientes elementos:

- **Base de datos**

A la base de datos del laboratorio se deberán añadir las nuevas tablas que se han creado para la aplicación web. Los pasos para realizar esta adición son los siguientes:

1. Copiar el script *web.sql* a la carpeta donde se encuentran el resto de scripts SQL del laboratorio.

Ruta: RLF_Web/scripts/web.sql.
2. Lanzar *mysql-query-browser*.
3. Cargar y ejecutar el script *web.sql* sobre la base de datos del laboratorio.

➤ Herramientas

Las nuevas herramientas que se han creado para la aplicación deben ser añadidas al laboratorio. Los pasos para añadirlas son los siguientes:

1. Copiar las carpetas *RLF_Chart* y *RLF_Cam* a la carpeta donde se encuentran el resto de herramientas del laboratorio.

Ruta: *RLF_Web/tools/RLF_Cam*

RLF_Web/tools/RLF_Chart

2. Abrir *rlfcam.xml* y *rlfchart.xml*. Cambiar el atributo *path* por la ruta actual.

Ruta: *RLF_Web/tools/RLF_Cam/rlfcam.xml*

RLF_Web/tools/RLF_Chart/rlfchart.xml.

3. Para garantizar que la transmisión del video del motor tenga audio, se deben seguir las instrucciones de la guía http://en.wikibooks.org/wiki/Configuring_Sound_on_Linux/HW_Address. En base al resultado obtenido, modificar el comando de ejecución del *VLC Player* en el fichero de ejecución *rlfcam*.

Ruta: *RLF_Web/tools/RLF_Cam/rlfcam*

4. Modificar las constantes *ip*, *port* y *video_length* en el fichero *rlfcam.xml*.

5. Modificar la constante *iterations* en el fichero *rlfchart.xml*.

6. Lanzar la aplicación del laboratorio.

Comando: *java -jar <ruta>/lab.jar.*

7. Registrar las dos nuevas herramientas.

Comando: *java -jar <ruta>/labconsole.jar [-h <IP> -p <Puerto>] -user <Usuario> -pass <Contraseña> <registry> <Parámetros>*

8. Abrir los ficheros de ejecución *rlfcam* y *rlfchart*. Modificar la variable *KEY* con la *KEY* devuelta por la aplicación del laboratorio cuando se registró cada herramienta

Ruta: *RLF_Web/tools/RLF_Cam/rlfcam*

RLF_Web/tools/RLF_Chart/rlfchart

9. Armar el laboratorio.

Comando: *java -jar <ruta>/labconsole.jar [-h <IP> -p <Puerto>] -user <Usuario> -pass <Contraseña> <arm> <Parámetros>*

10. Instalar *SQLite*, dado que las herramientas utilizarán este programa para su base de datos propia.

➤ **VLC Player**

Este programa debe instalarse dado que la herramienta *RLF_Cam* lo usará para transmitir video del motor. Se puede descargar desde la página <http://www.videolan.org/vlc/download-ubuntu.html>.

Servidor

El servidor se ha desarrollado en la versión 10.04.4 de Ubuntu, debido a que este proyecto tiene como base un proyecto anterior que utilizaba esta versión. La ISO puede ser descargada en la página <http://old-releases.ubuntu.com/releases/10.04.4/ubuntu-10.04.4-desktop-i386.iso>. A

continuación se indican las instalaciones que se deben llevar a cabo:

- **Oracle Java 7 JDK y JRE:** se deben seguir las instrucciones de la guía localizada en la página <http://www.wikihow.com/Install-Oracle-Java-on-Ubuntu-Linux>.
- **NetBeans 7.3.1 para Linux:** esta versión tiene *GlassFish* incluido. Si se instala otra versión, se debe añadir un servidor *Glassfish* o comprobar que la versión instalada permite su uso.

Seguidamente se debe cargar y configurar el proyecto correspondiente a la parte del servidor en *NetBeans*. Los pasos son los siguientes:

1. Importar el proyecto *RLF* en *NetBeans*.

Ruta: *RLF_Web/project/RLF*

2. Añadir *Glassfish* como servidor del proyecto.
3. Abrir la clase *Database.java* y modificar el código:

```
/** Localización de la base de datos*/

public final static String DATABASE = "jdbc:mysql://<IP>:<PUERTO>/rlf";

/** Usuario de acceso a la base de datos. */

private static String USER = <USUARIO>;

/** Contraseña del usuario. */

private static String PASS = <CONTRASEÑA>;
```

4. Abrir la clase *RLFWS endpoint.java* y modificar el código:

Método readMessage

```
String dbUrl = "jdbc:mysql:// <IP>:<PUERTO>/rlf";

Class.forName("com.mysql.jdbc.Driver");

Connection con = DriverManager.getConnection (dbUrl, <USUARIO>, <CONTRASEÑA>);
```

Variables globales

```
private final static int RLFCAM =<ID de la herramienta RLF_Cam>;

private final static int RLFCHART =<ID de la herramienta RLF_Chart>;

private final static String HELPPATH =<Ruta del fichero de ayuda del proyecto>

private final static String TESTPATH =<Ruta del fichero de cuestionario del proyecto>

private final static int BOOKING_TIME=<Tiempo de reserve deseado>;
```

Los ID de las herramientas se corresponden con los asignados a éstas por el laboratorio cuando fueron registradas.

5. Ejecutar el proyecto *RLF*.

Glosario de términos y siglas

UC3M: Universidad Carlos III de Madrid.

Front-end: parte de un sistema encargada de interactuar directamente con el usuario.

PID: Proporcional integral derivativo.

Middleware: programa que provee asistencia a una aplicación para la comunicación e interacción con otras aplicaciones, redes, hardware, etc.

DARPA: Defense Advanced Research Projects Agency.

MIT: Massachusetts Institute of Technology.

WAN: Wide Area Network.

ARPANET: Advanced Research Projects Agency Network.

TCP: Transmission Control Protocol.

IP: identificador de un dispositivo dentro de una red con protocolo IP.

UDP: User Datagram Protocol.

LAN: Local Area Network.

Host: ordenador conectado a una red a la cual provee y de la cual utiliza servicios.

Peer-to-peer: red punto a punto (P2P).

PC: Personal Computer.

DNS: Domain Name System.

ATM: Asynchronous Transfer Mode.

SQL: Structured Query Language.

HTML: HyperText Markup Language.

CSS: Cascading Style Sheets.

Plugin: aplicación que se añade a otra para aportarle una nueva funcionalidad.

IDE: Integrated Development Environment.

Streaming: transmisión de contenido multimedia a través de una red, permitiendo que el contenido sea consumido a la misma vez que es descargado.

JSON: JavaScript Object Notation.

Web Storage: Servicio de HTML5 que permite a las páginas web almacenar datos en el navegador.

W3C: World Wide Web Consortium.

Bash: Bourne again shell. Intérprete de comandos para Linux.

DBMS: Database Management System.

BBDD: Base de Datos.

RU: Requisito de usuario.

CU: Caso de uso.

RS: Requisito de Software.

WebSocket Endpoint: Objeto que gestiona una conversación a través de Websockets.

Token: cadena de caracteres identificativa que autoriza a un determinado usuario a utilizar herramientas.

Pop-up: Ventana emergente.

PF: Prueba de Funcionalidad.

Ghz: Gigahercio.

GB: Gigabyte.

RAM: Random-Access Memory.

HD: Hard Drive.

SAI: Sistema de Alimentación Ininterrumpida.

Bibliografía

[1] Rodríguez Mecha, Carlos A. “Diseño y desarrollo de un prototipo de Framework para laboratorios remotos”. Consultado en 2013 y 2014.

[2] Brief History of the Internet. Consultado en enero del 2014.

<http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet#History>

[3] Internet History. Consultado en enero del 2014.

<http://www.internethalloffame.org/internet-history/timeline>

[4] Internet. Consultado en enero del 2014.

www.uned.es/dpto-eeyc/asignaturas/653060/Internet.doc

[5] ISC Domain Survey. Consultado en enero del 2014.

<https://www.isc.org/services/survey/>

[6] A Timeline of Database History. Consultado en enero del 2014.

<http://quickbase.intuit.com/articles/timeline-of-database-history>

[7] A Brief History of Database Systems. Consultado en enero del 2014.

http://www.comphist.org/computing_history/new_page_9.htm

[8] Página oficial de NetBeans. Consultado en enero del 2014.

<https://netbeans.org/welcome.html>

[9] Netbeans vs Eclipse: An IDE comparison. Consultado en enero del 2014.

<http://software-talk.org/blog/2012/01/netbeans-vs-eclipse-an-ide-comparison/>

[10] Página oficial de Java. Consultado en enero del 2014.

http://www.java.com/en/download/faq/whatis_java.xml

http://www.java.com/es/download/whatis_java.jsp

<http://www.java.com/es/about/>

[11] Mozilla Developer Network: Javascript. Consultado en enero del 2014.

<https://developer.mozilla.org/es/docs/JavaScript>

https://developer.mozilla.org/es/docs/JavaScript/Acerca_de_JavaScript

[12] Página oficial de jQuery. Consultado en enero del 2014.

<http://jquery.com/>

[13] Introducción a JSON. Consultado en enero del 2014.

<http://www.json.org/json-es.html>

[14] Mozilla Developer Network: HTML. Consultado en enero del 2014.

<https://developer.mozilla.org/es/docs/Web/HTML>

[15] HTML5 Introduction. Consultado en enero del 2014.

http://www.w3schools.com/html/html5_intro.asp

[16] Mozilla Developer Network: CSS. Consultado en enero del 2014.

<https://developer.mozilla.org/es/docs/CSS>

[17] Cascading Style Sheets. Consultado en enero del 2014.

<http://www.w3.org/Style/CSS/>

[18] Scripts básicos para bash. Consultado en enero del 2014.

http://structio.sourceforge.net/guias/AA_Linux_colegio/scripts-basicos-para-bash.html

[19] GlassFish Server - Roadmap. Consultado en enero del 2014.

<https://glassfish.java.net/roadmap.html>

[20] Página oficial de MySQL. Consultado en enero del 2014.

<http://www.mysql.com/about/>

[21] Página oficial de SQLite. Consultado en enero del 2014.

<http://www.sqlite.org/about.html>

[22] The Java Tutorials: What Is a Socket? Consultado en enero del 2014.

<http://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>

[23] Socket. Consultado en enero del 2014.

<http://www.techterms.com/definition/socket>

[24] Mozilla Developer Network: WebSockets. Consultado en enero del 2014.

<https://developer.mozilla.org/en-US/docs/WebSockets>

[25] WebSocket.org: What is WebSocket? Consultado en enero del 2014.

<http://www.websocket.org/>

[26] About HTML5 WebSockets. Consultado en enero del 2014.

<http://www.websocket.org/aboutwebsocket.html>

[27] UML Designer User Guide. Consultado en enero del 2014.

<https://github.com/ObeoNetwork/UML-Modeling/blob/master/plugins/org.obeonetwork.dsl.uml2.design.doc/doc/UML%20Designer%20User%20Guide.textile>